A Close–to–Capacity Dirty Paper Coding Scheme

Uri Erez and Stephan ten Brink

Abstract—**The "writing on dirty paper"–channel model offers an information theoretic framework for precoding techniques for canceling arbitrary interference known at the transmitter. Using lattice strategies and MMSE scaling, lossless precoding is theoretically possible at any signal to noise–ratio. Following this approach, we design an end-to-end coding realization of a system materializing a significant portion of the promised gains. We employ vector quantization in combination with iterative decoding of capacity–approaching codes to achieve more than 2dB improvement over the best scalar quantization scheme. Code design is done using the EXIT chart technique.**

Index Terms: Interference cancellation, dirty paper coding, MMSE estimation, vector quantization, iterative decoding, repeat–accumulate codes, mutual information, EXIT chart.

I. Introduction

It has recently been shown [1] that an information theoretic framework for the study of efficient known interference cancellation (precoding) techniques may be found in Costa's "Writing on dirty paper" [2]. The (generalized) Dirty Paper Channel (DPC) model is depicted in Fig. 1. The received signal is

$$Y = X + S + N \tag{1}$$

where S is arbitrary interference known at the transmitter (non-causally), N is a statistically independent Gaussian random variable with variance P_N, and P_X is the power of the transmitted signal.

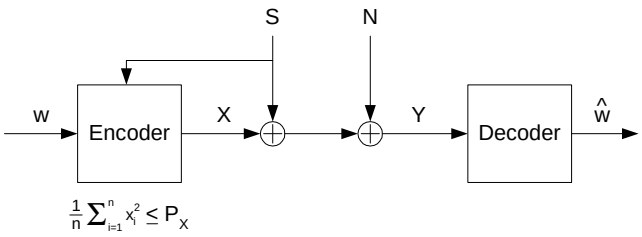

Fig. 1. The generalized Costa channel.

If the interference S was known at the receiver one could subtract it off the received signal leading back to an interference–free AWGN channel, and thus the interference would not pose a problem. One could similarly attempt to pre-subtract the interference at the transmitter, i.e., transmit $X' = X - S$. The received signal would then be $Y' = X' + S + N = X - S + S + N = X + N$, eliminating the interference. However, the problem with this naïve approach stems from the power constraint: The average transmit power would be $E[X'^2] = E[X^2] + E[S^2]$ (X, S assumed to be independent). As the interference may be arbitrarily strong, this would entail a severe power penalty and hence a reduced transmission rate. Nonetheless, in [2] Costa proved[1] that for Gaussian S and N, the capacity is equal to $\frac{1}{2}\log_2(1 + P_X/P_N)$ and hence the interference S does not incur any loss in capacity.

Costa proved his result using the general formula by Gelfand and Pinsker [3] for the capacity of channels with side information known at the transmitter. The paper did not address the relevance of the results to common communication problems and initially did not draw much attention, with the notable exception of Willems [4]. In the last few years, however, there have been a number of works that pointed out the connection of the DPC model to important communication problems. The connection to the problem of information embedding and digital watermarking was made in [5], [6] and [7]. In [1] the connection of the DPC model to precoding for interference cancellation was established, and Costa's result was extended to arbitrary interference, deterministic or random.

Recently, there has been considerable research studying the application of dirty paper coding to broadcast over multiple-input multiple-output (MIMO) channels, initiated in [8] and carried further in [9], [10], [11].

In such systems, in the eyes of a given user, the signals sent to other users act as interference. Since the users are physically separated, joint decoding is precluded. Furthermore, as the channel is not degraded, successive cancellation techniques at the receiver side are not applicable[2]. Nonetheless, as all signals are known to the transmitter successive "dirty paper" cancellation may be used in transmission after adequate linear preprocessing. These developments motivate finding realizable DPC coding techniques.

Willems [4] made the first step in suggesting schemes for coding for the dirty paper channel (for causally known interference), based on quantization of the interference. In [1] it was shown that the full capacity may be achieved using a scheme based on multidimensional lattice quantization and MMSE scaling. Related schemes were developed in the context of information embedding in [6] and [7]. In [12] a realization of the necessary lattice transmission scheme based on trellis shaping [13], [14] and "syndrome dilution" was proposed. Other approaches to designing multidimensional lattice-based dirty paper schemes were proposed in [15]. In this work we extend the approach of [12] by employing capacity–approaching codes and using iterative detection and decoding. We design a complete end-to-end dirty paper transmission system which attains a significant portion of the promised gains.

The paper is organized as follows. Section II reviews the lattice precoding approach of [1]. Section III describes coding for the one–dimensional (scalar) case. This will serve as a baseline

U. Erez is with EECS, MIT, Cambridge, MA. E-mail: uri@allegro.mit.edu. S. ten Brink is with Realtek, Irvine, CA. E-mail: stenbrink@realtek-us.com.

Part of this work has been presented at the *41st Ann. Allerton Conf. on Commun., Control, and Computing, Monticello, IL, Oct. 2003*, and the *Int. Symp. on Inform. Theory, Chicago, IL, Jun. 2004*.

The work of U. Erez has been supported by MIT's Deshpande Center for Technological Innovation.

[1] The model was proposed by T. Cover who also conjectured its capacity.

[2] One could of course impose a successive cancellation strategy at the receiver end but at signifi cant loss.

reference for our main results. Section IV provides additional background and an overview of the system designed. The main results of this work are reported in Sections V–VII. A brief outlook on further research is given in Section VIII, and a summary in Section IX.

II. Lattice precoding

We review the lattice precoding approach proposed in [1]. The key ingredients of the scheme are the use of lattice coding and decoding coupled with MMSE scaling. For a detailed account see [1], [16], [17]. We first briefly introduce lattices and then review the transmission scheme.

A. Lattices

A lattice Λ is a discrete subgroup of the Euclidean space $\mathbb{R}^n$ with the ordinary vector addition operation. Thus, if λ_1, λ_2 are in Λ, it follows that their sum and difference are also in Λ. A coset of Λ in $\mathbb{R}^n$ is any translated version of it, i.e., the set $\mathbf{x}+\Lambda$ is a coset of Λ for any $\mathbf{x} \in \mathbb{R}^n$. The fundamental Voronoi region of $\Lambda \subset \mathbb{R}^n$, denoted by $\mathcal{V}$, is the set of minimum Euclidean norm coset representatives of the cosets of Λ. Every $\mathbf{x} \in \mathbb{R}^n$ can be uniquely written as $\mathbf{x} = \lambda + \mathbf{r}$ with $\lambda \in \Lambda$, $r \in \mathcal{V}$, where $\lambda = Q_{\mathcal{V}}(\mathbf{x})$ is a nearest neighbor of $\mathbf{x}$ in Λ, and $\mathbf{r} = \mathbf{x} \bmod \Lambda$ is the apparent error $\mathbf{x} - Q_{\mathcal{V}}(\mathbf{x})$. We may thus write $\mathbb{R}^n = \Lambda + \mathcal{V}$ and $\mathcal{V} = \mathbb{R}^n \bmod \Lambda$. For a comprehensive introduction to lattices we refer the reader to [18].

A simple family of lattices that will repeatedly appear in the sequel is that of cubic lattices. In one dimension it is simply the set of integers $\mathbb{Z}$, or any scaled version of it $a\mathbb{Z}$. An n-dimensional cubic lattice is an orthogonal transformation of $\mathbb{Z}^n$, the n-fold Cartesian product of $\mathbb{Z}$. To be consistent in notation, we take $x \bmod \mathbb{Z}$ to denote reducing x to the interval $(-1/2, 1/2]$, the fundamental Voronoi region of $\mathbb{Z}$. Note that this differs from the usual convention where the interval is taken to be $[0, 1)$.

We denote by $|\mathcal{V}|$ the volume of a Voronoi region. The averaged (per dimension) second moment of (the fundamental Voronoi region of) a lattice is

$$P(\Lambda) = \frac{1}{n|\mathcal{V}|}\int_{\mathcal{V}} \|\mathbf{x}\|^2 d\mathbf{x}. \tag{2}$$

The volume $|\mathcal{V}|$ and averaged second moment $P(\Lambda)$ of a lattice are related by the normalized second moment of the lattice $G(\Lambda)$, defined by

$$G(\Lambda) = \frac{P(\Lambda)}{|\mathcal{V}|^{2/n}}. \tag{3}$$

Thus, if the Voronoi region has unit volume, $G(\Lambda)$ is just the averaged second moment of $\mathcal{V}$. The definition ensures that $G(\Lambda)$ is invariant under scaling (and isometry). We have $G(T\,\Lambda) = G(\Lambda)$ where T is any orthogonal transformation. Also note that for a hypercube of any dimension we have

$$G(\mathbb{Z}^n) = \int_{-1/2}^{1/2} x^2 dx = \frac{1}{12}. \tag{4}$$

It is easy to see that for any dimension n the region that has the smallest normalized second moment (defined in the obvious manner, similarly to (3)) is the n-sphere. We also have

$$G(n\text{−sphere}) > \frac{1}{2\pi e} \approx \frac{1}{17}. \tag{5}$$

and $G(n\text{−sphere}) \to \frac{1}{2\pi e}$ as $n \to \infty$. It is known [19] that there exist good lattices for shaping Λ_n in the sense that $G(\Lambda_n) \to \frac{1}{2\pi e}$. The *shaping gain* $g_s(\Lambda)$ of a lattice Λ is defined as

$$g_s(\Lambda)|_{\mathrm{dB}} = 10\log_{10}\frac{G\left(\mathbb{Z}^n\right)}{G\left(\Lambda\right)} = 10\log_{10}\frac{1}{12G\left(\Lambda\right)}. \tag{6}$$

It quantifies the gain in using $\mathcal{V}$ for shaping w.r.t. to a hypercube (no shaping). That is, it measures how much more power is needed when using a uniform cubic input distribution rather than a distribution uniform over the Voronoi region $\mathcal{V}$. Therefore, the ultimate shaping gain with respect to a cubic region is

$$g_s(\Lambda)|_{\mathrm{dB}}\,(\text{optimal shaping}) = 10\log_{10}\frac{2\pi e}{12} \approx 1.53\text{dB}. \tag{7}$$

B. Communication over modulo–lattice channels

Let Λ denote an n-dimensional lattice with fundamental Voronoi region $\mathcal{V}$ having averaged second moment $P(\Lambda) = P_X$. Also let $\mathbf{U} \sim \mathrm{Unif}(\mathcal{V})$, that is, $\mathbf{U}$ is a random variable (dither) uniformly distributed over $\mathcal{V}$. The scheme is given by,

- *Transmitter:* The input alphabet is restricted to $\mathcal{V}$. For any $\mathbf{v} \in \mathcal{V}$, the encoder sends:

$$\mathbf{X} = [\mathbf{v} - \alpha\mathbf{S} - \mathbf{U}] \bmod \Lambda. \tag{8}$$

- *Receiver:* The receiver computes

$$\mathbf{Y}' = [\alpha\mathbf{Y} + \mathbf{U}] \bmod \Lambda. \tag{9}$$

The resulting channel is a mod-Λ additive noise channel described by the following lemma:

Mod Λ–*channel* [1]: *The channel from* $\mathbf{v}$ *to* $\mathbf{Y}'$ *defined by (1),(8) and (9) is equivalent in distribution to the mod–*Λ *channel*

$$\mathbf{Y}' = [\mathbf{v} + \mathbf{N}'] \bmod \Lambda \tag{10}$$

with

$$\mathbf{N}' = [(1-\alpha)\mathbf{U} + \alpha\mathbf{N}] \bmod \Lambda. \tag{11}$$

Note that due to the dither, $\mathbf{X}$ is uniformly distributed over $\mathcal{V}$, independent of $\mathbf{v}$, and has power $E\left[\|\mathbf{X}\|^2\right] = P_X$. The mutual information of the channel is maximized by a uniform input $\mathbf{V} \sim \mathrm{Unif}(\mathcal{V})$, giving

$$\begin{aligned}
\frac{1}{n}I(\mathbf{V};\mathbf{Y}') &= \frac{1}{n}h(\mathbf{Y}') - \frac{1}{n}h(\mathbf{N}') \\
&= \frac{1}{n}\log_2|\mathcal{V}| - \frac{1}{n}h(\mathbf{N}') \\
&= \frac{1}{2}\log_2\frac{P_X}{G(\Lambda)} - \frac{1}{n}h(\mathbf{N}') \\
&= \frac{1}{2}\log_2 2\pi e P_X - \frac{1}{n}h(\mathbf{N}') \\
&- \frac{1}{2}\log_2 2\pi e G(\Lambda).
\end{aligned} \tag{12}$$

We next bound $h(\mathbf{N}')$ as follows

$$\begin{aligned}\frac{1}{n}h(\mathbf{N}') &\leq \frac{1}{n}h((1-\alpha)\mathbf{U}+\alpha\mathbf{N}) \\ &\leq \frac{1}{n}\log_2\Big(2\pi e E\left[\|(1-\alpha)\mathbf{U}+\alpha\mathbf{N}\|^2\right]\Big) \quad (13)\end{aligned}$$

where the first inequality follows since the modulo operation can only decrease the entropy and the second inequality follows since for a given second moment a Gaussian random vector has the greatest entropy.

We further have

$$\begin{aligned}\frac{1}{n}E\left[\|(1-\alpha)\mathbf{U}+\alpha\mathbf{N}\|^2\right] &= \frac{1}{n}\sum_{i=1}^{n}\Big((1-\alpha)^2E[U_i^2] \quad (14)\\ &+ \alpha^2E[N_i^2] \quad (15)\\ &+ (1-\alpha)\alpha E[U_iN_i]\Big) \quad (16)\\ &= \frac{1}{n}E[\|\mathbf{U}\|^2]+\alpha^2\mathrm{Var}\,[N] \quad (17)\\ &= (1-\alpha)^2P_X+\alpha^2P_N. \quad (18)\end{aligned}$$

where (17) follows since $E[N_i]=0$.

We next choose α to minimize (18), resulting in[3] $\alpha^* = \frac{P_X}{P_X+P_N} = \frac{\mathrm{SNR}}{1+\mathrm{SNR}}$, where $\mathrm{SNR}=P_X/P_N$. With this choice we have

$$\begin{aligned}\frac{1}{n}E\left[\|(1-\alpha^*)\mathbf{U}+\alpha^*\mathbf{N}\|^2\right] &= (1-\alpha^*)^2P_X+\alpha^{*2}P_N\\ &= \frac{P_NP_X}{P_N+P_X}\\ &= \alpha^*P_N. \quad (19)\end{aligned}$$

Combining (12), (13) and (19) we obtain the following lower bound on the achievable rate as a function of $G(\Lambda)$,

$$I(\mathbf{V};\mathbf{Y}') \geq \frac{1}{2}\log_2(1+\mathrm{SNR}) - \frac{1}{2}\log_2 2\pi e G(\Lambda). \quad (20)$$

In principle, for a given lattice Λ, the gap to capacity of a precoding system may be made smaller than $\frac{1}{2}\log_2 2\pi eG(\Lambda)$. For optimal lattices for shaping we have $G(\Lambda)\to\frac{1}{2\pi e}$ and thus the gap goes to zero.

Fig. 2 depicts the lower bound (20) on the achievable rate in bits per two dimensions as a function of E_b/N_0, for a given shaping gain. With (6) we compute

$$bound = \log_2(1+\mathrm{SNR}) - \log_2\left(\frac{2\pi e}{12}\cdot 10^{-\frac{g_s(\Lambda)|_{\mathrm{dB}}}{10}}\right)$$

and plot the parametric curve $\big(10\log_{10}\frac{\mathrm{SNR}}{bound}, bound\big)$.

For one–dimensional Λ the lattice precoding scheme is based simply on scalar quantization (SQ) and is an extension of Tomlinson-Harashima precoding [20], [21] incorporating MMSE scaling (scaling by α). For this case, the achievable mutual information of the mod-Λ channel (10) may be easily computed and is depicted in the Fig. 2. While the gap to capacity of a scalar system is $10\log_{10}(2\pi e/12)\approx 1.53$dB at high SNR, the lowest possible E_b/N_0–operating point is at 2.4dB. This means that the gap to capacity approaches 4dB at zero spectral efficiency (see Fig 2). For this reason we concentrate our efforts on the low SNR regime.

Thus, shaping plays a very different role in DPC coding when compared to coding for an interference–free AWGN channel. At high SNR the shaping gain is 1.53dB in both cases. However, for an AWGN channel shaping becomes unnecessary at low SNR, while the importance of shaping grows in DPC coding as the SNR *decreases*. This central role of shaping at low spectral efficiencies poses a challenge in terms of coding. Practical and effective methods for shaping have been developed by [13], [22]. While we follow the approach of trellis shaping [13] to generate effective shaping codes, the architecture of trellis precoding techniques as in [14] (designed for high spectral efficiencies) is not applicable and it is necessary to develop new schemes.

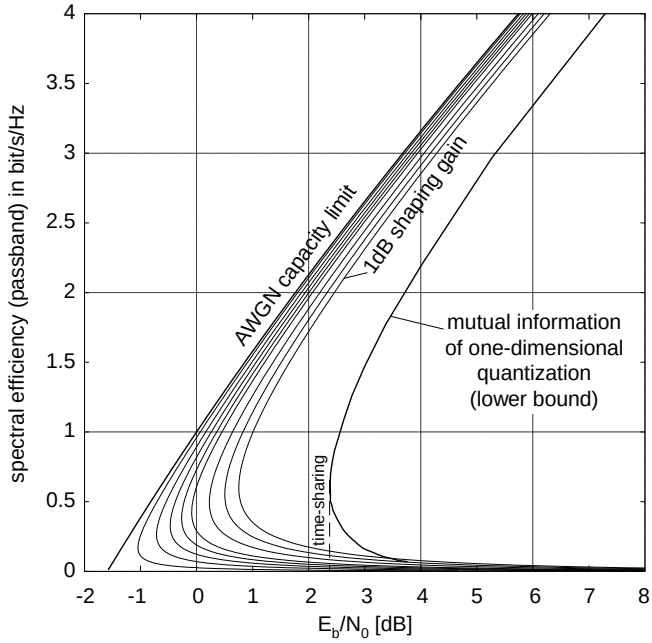

Fig. 2. Lower bound (20) on achievable rates for various values of $g_s(\Lambda)$. From left to right: AWGN capacity limit; lower bound for shaping gains $1.5, 1.45, 1.4, 1.35, 1.3, 1.2, 1.1$ and 1.0dB; mutual information of one–dimensional scheme.

III. One–dimensional (Scalar) Quantization

We first describe a one–dimensional (scalar) lattice transmission system, as depicted in Fig. 3. See [23] for a similar scalar system.

The effective noise channel (10) takes the form

$$Y' = [v + \underbrace{(1-\alpha)\,U}_{\substack{\text{uniform in interval}\\(-A(1-\alpha),A(1-\alpha)]}} + \underbrace{\alpha N}_{\substack{\text{Gaussian}\\N\left(0,\alpha^2P_N\right)}}\ ] \bmod 2A\,\mathbb{Z}.$$

For simplicity, we use $A=2$ for the remainder of this paper, with modulo interval $(-2,2]$. Computing the mutual information $I(V;Y')$ for different values of α by Monte-Carlo simulation (assuming BPSK transmission $V=\pm 1$) provides the mutual information limits shown in Fig. 5. The E_b/N_0–value is

[3]This choice of α goes back to Costa's paper [2] and is related to MMSE estimation, see [17].

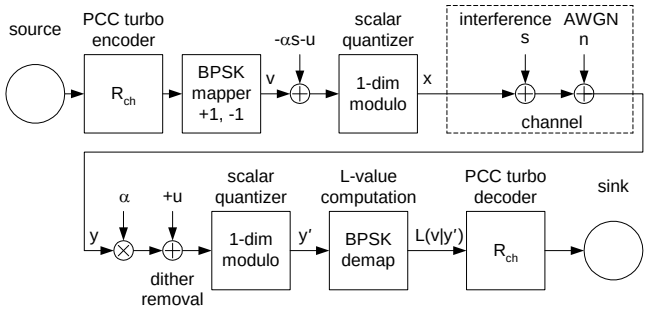

Fig. 3. Dirty paper coding with turbo codes and a scalar quantizer (SQ).

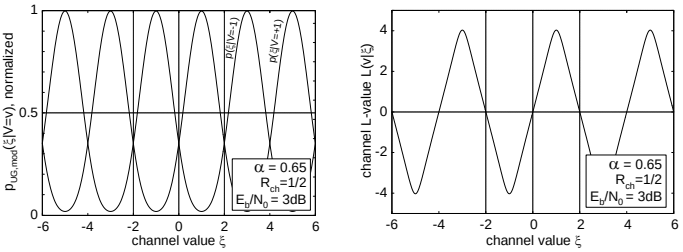

Fig. 4. Left: Conditional probability density functions (convolution of Gaussian and uniform densities) after scalar quantization. Right: Corresponding log–likelihood ratio values; modulo interval $(-2, 2]$.

based on the actual output power P_X. As X is uniformly distributed in $(-2, 2]$, we have

$$P_X = P_{uni,(-2,2]} = \frac{1}{4}\int_{-2}^{2} \xi^2 \, d\xi = \frac{4}{3} \approx 1.249\text{dB} \qquad (21)$$

Coding for this channel is essentially not much different than for an AWGN channel. We use an off–the–shelf parallel concatenated (turbo) code (PCC, [24]), and compute an appropriate soft–input metric (L–values [25]) to the turbo decoder based on a one–dimensional modulo.

A. Log-likelihood ratio values based on modulo metric

With $\sigma^2 = P_N$ the PDF of the i.i.d. Gaussian noise αN on the channel writes as

$$p_G(\xi) = \frac{1}{\alpha^2\sigma^2\sqrt{2\pi}} \cdot \exp\left[-\frac{\xi^2}{2\alpha^2\sigma^2}\right]. \qquad (22)$$

Correspondingly, the PDF of $(1-\alpha)U$ is

$$p_U(\xi) = \begin{cases} \frac{1}{4(1-\alpha)} & ; \; -2(1-\alpha) \le \xi \le 2(1-\alpha) \\ 0 & \text{else} \end{cases}. \qquad (23)$$

The convolution $p_{UG}(\xi) = p_U(\xi) * p_G(\xi)$ yields the PDF of the sum $(1-\alpha)U + \alpha N$ as

$$p_{UG}(\xi) = \begin{cases} \frac{\text{erf}\left(\frac{\xi+2\cdot(1-\alpha)}{\alpha\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{\xi-2\cdot(1-\alpha)}{\alpha\sigma\sqrt{2}}\right)}{8\cdot(1-\alpha)}; & 0 < \alpha < 1 \\ p_G(\xi); & \alpha = 1 \end{cases}. \qquad (24)$$

After the one–dimensional modulo at the receiver, only a few neighboring modulo–intervals need to be considered in practice (correponding to k running form -3 to 3 in (25)). The L–value for BPSK demapping computes as

$$L(v\,|y') \approx \ln \frac{\sum_{k=-3}^{3} p_{UG}(y' - 1 + 4k)}{\sum_{k=-3}^{3} p_{UG}(y' + 1 + 4k)}. \qquad (25)$$

Fig. 4 depicts a typical effective modulo–noise channel and the respective L–values.

B. Simulation results using turbo codes

As our focus is on low SNR, we chose our target operating spectral efficiency (passband) to be 1bit/s/Hz. We apply a PCC of rate $R_{ch} = 1/2$ and BPSK modulation per dimension. The code is of memory 4, and has generator polynomials 037_8 (feedback) and 021_8 (feedforward). With $\alpha = 0.65$, we obtain a turbo cliff at about 3dB (length $K = 10^5$ systematic bits, 20 iterations) which is just about 0.4dB from the performance predicted by the mutual information limits of the scalar quantizer, and 3dB away from the AWGN capacity limit (see Fig. 5). Similarly, for a spectral efficiency of 0.667bit/s/Hz, we use a PCC of rate $R_{ch} = 1/3$, memory 4, and polynomials 025_8 (feedback), 037_8 (feedforward). Setting $\alpha = 0.55$, we get the turbo cliff at about 2.8dB which is 0.4dB from the mutual information limit, and 3.4dB away from the respective AWGN capacity limit.

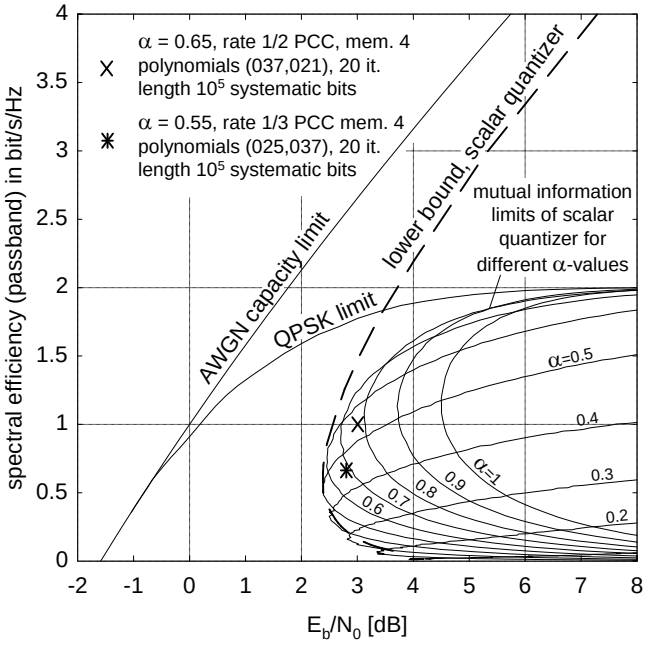

Fig. 5. Mutual information limits of SQ and BPSK/dimension for different α–values (in steps of 0.1) in the spectral efficiency chart; PCC turbo cliff positions at 3dB (1bit/s/Hz) and 2.8dB (0.667bit/s/Hz) respectively.

IV. Multidimensional (Vector) Quantization

A. Background: Obtaining lattices from linear codes

Consider the lattice transmission scheme of Section II. The modulo operations performed at both transmission ends, i.e., those in (8) and (9), mean that we may equivalently view a message selection as specifying a *coset* $\mathbf{v} + \Lambda$. The actual

transmitted signal is the difference between $\alpha\mathbf{S}+\mathbf{U}$ and the nearest point of the coset. As $\mathbf{S}$ is arbitrary (unbounded) this means that we have to search over the infinite lattice. Similarly, while we may closely approximate the "unfolded" effective noise $(1-\alpha)\mathbf{U}+\alpha\mathbf{N}$ as Gaussian noise of variance $(1-\alpha)^2P_X+\alpha^2P_N$ the modulo operation (folding) at the receiver means that we have to compute the metrics

$$\sum_{\lambda\in\Lambda}\exp\left[-\frac{\|\alpha\mathbf{Y}+\mathbf{U}-\mathbf{v}+\lambda\|^2}{2\left((1-\alpha)^2P_X+\alpha^2P_N\right)}\right] \tag{26}$$

where $\mathbf{v}$ is a hypothesized codeword. Again, this involves a summation over the infinite lattice which in effect performs the modulo operation specified in (9). While this poses no real problem in the scalar case, as scalar quantization is a very simple operation, it is an issue that has to be addressed when high dimensional lattices are used.

Fortunately, a standard method for constructing lattices from linear codes, i.e., *Construction A* (see [26]), yields lattices that are also periodic in the cubic lattice $q\mathbb{Z}^n$. Furthermore, lattices which are optimal for shaping, i.e., having $\log_2 2\pi eG(\Lambda)$ as small as desired, may be obtained having this structure (although one would have to use non–binary codes). This reduces the search to that of first performing *one*–dimensional (scalar) quantization, and then performing a search over the finite set of coset representatives of the quotient group $\Lambda/q\mathbb{Z}^n$. The separation of the search into these two stages is done in trellis precoding in [14] and in the context of DPC coding in [12]. Fig. 6 illustrates the construction by example over $\mathbb{Z}_{11}$.

Example: We take block length $n=2$ and field $\mathbb{Z}_{11}$. We use a rate $1/2$ block code ($k=1$) given by the generating matrix (vector) $G=[2,3]$ so that the code $\mathcal{C}$ is given by

$$\mathcal{C}=\{x\cdot[2,3] \bmod 11 \ : \ x\in\mathbb{Z}_{11}\}$$

We embed the code "as is" in Euclidean space as depicted in Fig. 6 (left).

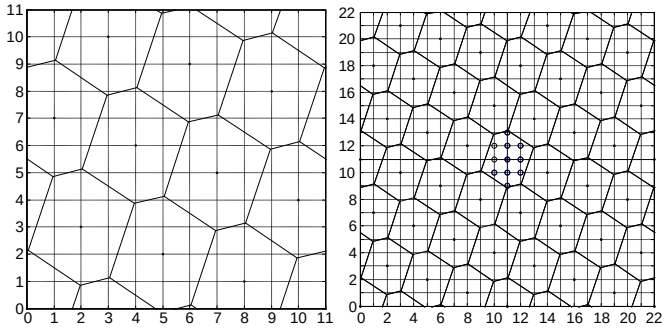

Fig. 6. Construction A: Linear code is embedded in $\mathbb{R}^2$ (left), then space is tessellated (right)

Then, using this "finite lattice", we tessellate the whole of $\mathbb{R}^2$ giving the lattice

$$\Lambda=\mathcal{C}+11\mathbb{Z}^2\,.$$

The eleven points contained in the fundamental Voronoi region serve as coset representatives and correspond to the choice of the vector $\mathbf{v}$ in Section II. These coset representatives give us an uncoded but shaped constellation. The next step is to choose a code, i.e., use a subset of these coset leaders and map the messages onto them.

B. "Construction A" and convolutional codes

In the designed system the linear codes used are binary convolutional codes as is often done in practice. Following [12] we use sign–bit trellis shaping [13] to generate the shaping lattice. We briefly review the encoding scheme. We then outline the decoding process which is detailed in the next section.

For simplicity consider a rate $R_{VQ}=1/2$ (where the subscript VQ stands for vector quantization) convolutional code[4] corresponding to an encoder with generating polynomials $(g_1(D),g_2(D))$. The resulting code is given by the pair of all possible output sequences

$$c_i(D)=g_i(D)u(D)\quad i=1,2 \tag{27}$$

where $u(D)$ ranges over all input sequences. We next combine the two output codewords into one code sequence $c(D)$ by interlacing them

$$c_k=\begin{cases} c_{1,k/2} & k \text{ even} \\ c_{2,(k-1)/2} & k \text{ odd}\end{cases}. \tag{28}$$

Denote the resulting code by $\mathcal{C}$. We note that as transmission is over the period $1,\ldots,n$ we are assuming that u_k is zero for $k\leq 0$ and appropriately terminated. Thus, from the convolutional code we obtain a block code of length n.

As in the example above we obtain a lattice in $\mathbb{R}^n$ by applying Construction A. Thus,

$$\Lambda=\mathcal{C}+2\mathbb{Z}^n. \tag{29}$$

That is, any point of the lattice can be obtained by adding a sequence of even integer components to a codeword $\mathbf{c}\in\mathcal{C}$. Note that by construction, applying a one–dimensional quantizer to Λ results in $\mathcal{C}$, i.e.,

$$\Lambda \bmod 2\mathbb{Z}^n=\mathcal{C}. \tag{30}$$

C. Vector quantization at the transmitter

We now describe how quantization and the modulo operation with respect to Λ may be performed. Consider a sequence $\mathbf{x}\in\mathbb{R}^n$. We wish to find $\mathbf{x} \bmod \Lambda=\mathbf{x}-Q_\Lambda(\mathbf{x})$. That is, we wish to find the nearest point of Λ to $\mathbf{x}$ in Euclidean sense and take the difference between the two. Due to the periodicity of the lattice in $2\mathbb{Z}^n$ we have

$$\mathbf{x} \bmod \Lambda=[\mathbf{x} \bmod 2\mathbb{Z}^n] \bmod \Lambda. \tag{31}$$

Thus, we may first perform one–dimensional quantization and then search over a finite set. Denote the output of this first stage quantization by $\mathbf{x}'=\mathbf{x} \bmod 2\mathbb{Z}^n$. Note that $x'_k\in(-1,1]$ for all k. Now define the modulo Euclidean distance between $\mathbf{x}'$ and the code $\mathcal{C}$ to be

$$d^2(\mathbf{x}',\mathcal{C})=\min_{\mathbf{c}\in\mathcal{C}}\left\{\sum_{k=1}^{n}\left|[x'_k-c_k] \bmod 2\mathbb{Z}\right|^2\right\}. \tag{32}$$

[4]Only rate $1/2$ codes are in fact used in the system designed. We use boldface to denote sequences or vectors as in $\mathbf{x}$, x_k to denote its entries in the time domain and $x(D)$ to denote its formal D-transform.

A moment of reflection confirms that indeed

$$\min_{\lambda\in\Lambda} \|\mathbf{x}-\lambda\|^2 = d^2(\mathbf{x}',\mathcal{C}). \tag{33}$$

The minimization in (32) is performed using the Viterbi algorithm applied to the trellis defined by the convolutional code. See the literature on trellis shaping (e.g., [13]) for details.

We are now ready to map an information sequence into the Voronoi region $\mathcal{V}$. We define the *coset leader* as the unique member of the coset lying in the fundamental Voronoi region, i.e., the member of the coset having smallest Euclidean norm. Thus, any integer sequence $u(D)$ is mapped to its coset leader by reducing it mod-Λ. The set of coset leaders is given by

$$\mathcal{L} = \mathbb{Z}^n \bmod \Lambda. \tag{34}$$

Therefore, we may obtain the set $\mathcal{L}$ by reducing all binary sequences modulo the shaping lattice. We view $\mathcal{L}$ as an uncoded but shaped "constellation". It is easy to see that while an unconstrained binary sequence represents one bit of information per symbol, the set of coset leaders represents (or in the nomenclature of [27], has an informativity of) $1-R_{VQ}$ bits per dimension[5]. For a more precise and comprehensive treatment we refer the reader to [13]. In our case we have $R_{VQ}=1/2$. Thus, if we use BPSK signaling we start out with a constellation of 0.5 bit/dimension in place of 1 bit/dimension, and we write informally

$$\mathbb{Z}_2^n \bmod \Lambda = \text{uncoded constellation of } 0.5 \text{ bit/dim}. \tag{35}$$

Note that the signal points are 0 and 1 in this notation. Nevertheless, the transmitted signal is zero mean. In fact, it is symmetric around the origin and has a truncated Gaussian–like shape in the interval $(-1,1]$. One may also use any shifted version of the constellation. Thus, if we used a standard BPSK constellation $\{-1/2,1/2\}$ and correspondingly shifted the shaping lattice, the result would be the same. Similarly, we may multiply both the lattice and the constellation with a constant factor A to obtain a more convenient representation. Indeed, in the simulations described in the sequel as well as in Section III we chose $A=2$ to have a BPSK constellation $\{-1,1\}$, and a transmitted signal in the interval $(-2,2]$.

To get a higher rate we may simply start with a larger constellation and reduce it modulo a multiple of Λ. For instance we may use a 4-PAM constellation and reduce it modulo 2Λ. This gives an effective constellation of $2-0.5=1.5$ bits per dimension and we may write

$$\mathbb{Z}_4^n \bmod 2\Lambda = \text{uncoded constellation of } 1.5 \text{ bit/dim}. \tag{36}$$

The modulo operation effects only the most significant bit (MSB). Hence this shaping technique is referred to as sign–bit shaping [13].

The actual operation of mapping information sequences to coset leaders is straightforward. It consists of taking an arbitrary information sequence and first upsampling it (in our case by a factor of two). This maps it to some member of a coset (with respect to Λ). As long as distinct information sequences are mapped to different cosets this operation is information preserving. Next, the resulting coset member is reduced modulo Λ to arrive at the coset leader. In the designed system the upsampler is taken to be a *repetition code*.

At this point we have arrived at a system that maps an arbitrary uncoded input sequence to an uncoded but shaped sequence. The next step is "diluting" the input bit stream by means of a channel code as proposed in [12]. Thus, the information bits are first passed through a channel code adding redundancy, then upsampled, mapping the sequence to a coset, and finally reduced mod-Λ to a good coset leader. The last step, as far as encoding is concerned, is to compute the difference between the scaled interference (after adding the dither) and the chosen coset leader sequence, reduce it modulo-Λ and send it over the channel. This encoding operation is schematically shown in Fig. 7[6]. ¿From the figure it is evident that there is no need to perform the first pair of modulo operations, and this step *is not implemented* in the system. However, it is helpful as a conceptual aid.

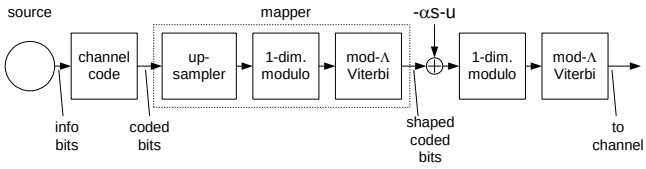

Fig. 7. Schematic description of dirty paper encoder.

For a rate $R_{VQ}=1/2$ VQ based on binary convolutional codes and sign–bit shaping, we measured shaping gains of 0.98dB (memory 2, polynomials $05_8, 07_8$) 1.13dB (memory 4, $023_8, 035_8$), 1.215dB (memory 6, $0133_8, 0171_8$) and 1.28dB (memory 8, $0561_8, 0753_8$), respectively. We shall use these convolutional codes for simulating vector quantizer schemes in Sections VI–VII. For comparison, the shaping gain of the 24–dimensional Leech lattice Λ_{24} is only 1.03dB [27], while being more complex to implement than a VQ based on a memory 2 convolutional code. The shaping gain computes as

$$g_s|_{\rm dB} = P_{uni,(-1,1]}\big|_{\rm dB} - P_X|_{\rm dB} + 10\log_{10} R_{VQ}$$

with $P_{uni,(-1,1]}$ computed similarly as in (21), and P_X being the power of the (truncated Gaussian–like) shaped signal at the output of the TX-VQ.

D. Vector quantization at the receiver

We next describe the decoder. It is here that we must depart from previous approaches. Conceptually, following the lattice decoding scheme described in Section II, we would like to apply a mod-Λ operation (after scaling by α), which could be done using a Viterbi algorithm. However, there are two drawbacks to this approach. First, as explained in the example, computing the density of the effective noise $\mathbf{N}'$ involves in itself a search over the lattice, and thus a front–end modulo operation at the receiver would be pointless. Furthermore, we are using a capacity approaching code (specifically, a repeat–accumulate code) based on iterative decoding. The decoder for the channel code requires

[5]Note that in [27] the informativity is defined in bits per *two* dimensions.

[6]The figure depicts the operation of a BPSK system. Some minor changes needed in the 4-PAM system are described in Section V.

knowledge of the posterior probabilities of the bits that were input to the mapper, i.e., to the combined upsampler and Viterbi module at the transmitter. However, applying a Viterbi algorithm as a front end of the receiver does not offer this, rather it performs hard sequence-wise detection, ignoring the coded nature of the input bits. Of course, theoretically one could perform a joint Viterbi detector that is based on the combined trellis of the channel and the shaping code. This however would miss the mark as then there is no point in using a concatenated channel code.

Rather, we replace the Viterbi decoder with a BCJR A Posteriori Probability (APP) decoder [28]. The BCJR supplies the initial APP values of the input bits to the channel decoder. Thus, it may be viewed as a bit-wise *quantization detector*. The channel decoder comes into play, refining the estimates of the bit probabilities using the redundancy in the coded bit stream. These are then fed back to the BCJR module and so we proceed in an iterative fashion until convergence. The general architecture is depicted in Fig. 8. This procedure will further be refined in the next section to reflect the specific structure of the RA codes employed. We note that the decoder is quite different from that

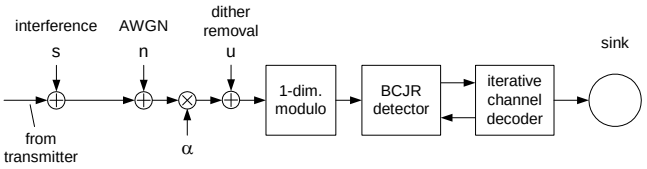

Fig. 8. Schematic description of dirty paper decoder.

used in combined trellis shaping and coding as proposed in [14]. In particular we do not use a "syndrome former" for decoding, thus avoiding the need for specific inverse syndrome formers at the transmitter. In our system the inverse syndrome former is simply replaced by an upsampler. Also, as we are aiming at low spectral efficiencies, there are no *uncoded bits*.

V. Designed system

A. End–to–end dirty paper coding link

The detailed system model is depicted in Fig. 9. It incorporates a check–biregular, repeat–irregular nonsystematic repeat–accumulate (RA) code [29], [30] concatenated with a trellis shaping code. The variable node decoder (VND) of the RA code is designed for iterative quantization detection and decoding using the EXIT chart technique. We note that, in principle, a low–density parity–check (LDPC) code could be used in a similar set–up, while applying the same code design steps as described in the later sections. However, irregular RA codes exhibit a linear encoding complexity, convenient for Monte–Carlo simulation.

The transmitter is a concatenation of a nonsystematic RA code, performing the "coset dilution", and a trellis shaping code (i.e. the vector quantizer). The RA encoder is composed of an outer mixture of repetition codes of different rates (variable nodes), an edge interleaver, and an inner mixture of single parity check codes of different rates (check nodes), followed by a memory one differential encoder (accumulator, ACC). *Inner*

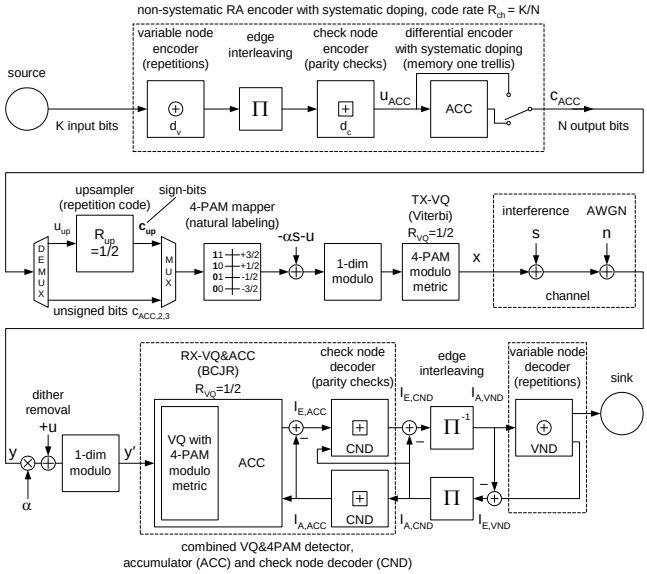

Fig. 9. Dirty paper coding with nonsystematic repeat-accumulate codes (using inner systematic doping) and a vector quantizer (VQ); iterative quantization and decoding.

systematic doping can be applied, that is, some of the coded bits of the accumulator output can be *substituted* by the corresponding systematic bits at the accumulator input. Code design is performed by appropriately choosing repetition and check node degree distributions. The encoded bits are grouped into triplets $(c_1, c_2, c_3)_{ACC}$ and demultiplexed into "upsampler" bits $u_{up} = c_{ACC,1}$ and unsigned bits $c_{ACC,2}, c_{ACC,3}$. The upsampler (replacing the inverse syndrome former in trellis shaping) has rate $R_{up} = 1 - R_{VQ} = 1/2$. The sign–bits $c_{up,1}, c_{up,2}$ generated by the upsampler, and the unsigned bits are mapped onto 4-PAM symbols using natural labeling. After adding the scaled interference and a uniformly distributed dither signal, the vector quantizer determines (using the Viterbi algorithm) the minimum energy sequence ("shaping"), and the quantization error vector $\mathbf{x}$ is transmitted over the communication channel. The output power of the (truncated) Gaussian–like signal is P_X per real dimension. We define E_s as the average energy per *complex* output symbol, i.e., $E_s = 2P_X$.

On the channel, white Gaussian noise is added, with double–sided noise power spectral density $P_N = N_0/2$ and zero mean. Interference is added. For 16-QAM (4-PAM per dimension) and $R_{VQ} = 1/2$, we have $E_s/N_0 = 2(1 + 0.5)R_{ch}E_b/N_0$. Thus, for simulation we set $P_N = E_s/(3R_{ch}2E_b/N_0)$.

At the receiver, MMSE α–scaling is applied, and the dither signal U is removed; a one–dimensional modulo is performed prior to passing the signal into a soft in/soft out vector quantizer which performs an *a posteriori* probability (APP) detection of the sign–bits and the unsigned bits respectively, using the BCJR algorithm [28] on an appropriately defined trellis structure. The vector quantizer, thus, can be viewed as an APP detector, computing extrinsic information on the sign–bits and unsigned bits respectively, which is forwarded to the RA decoder. The RA decoder is composed of an inner accumulator decoder (ACC), check node decoder (CND), and an outer variable node decoder (VND), which, in turn, provides *a priori* information for the

APP VQ detector to improve the quantization result ("iterative quantization and RA decoding"). The structure is close to the scheme presented in [30]. As we merge the APP vector quantizer with the inner accumulator decoder of the RA code, we obtain a variant of "trellis detection", similar to [31].

B. Joint accumulator and quantizer trellis processing

Fig. 10 aids in understanding the structure of the joint trellis processing over the accumulator trellis (memory $\nu_{ACC} = 1$), vector quantizer trellis (memory ν_{VQ}), upsampler, and modulo symbol metric based on two 4-PAM symbols per three hypothesized accumulator bits $(u_1, u_2, u_2)_{ACC}$.

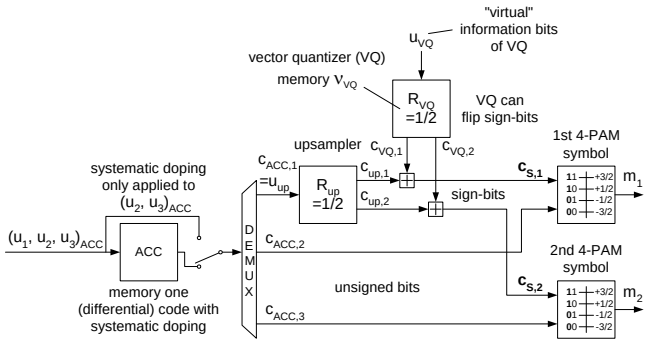

Fig. 10. Illustration of joint accumulator (using systematic doping), upsampler, vector quantizer and 4-PAM trellis processing.

One *trellis column* comprises $2^{\nu_{ACC}} = 2$ states of the accumulator, and $2^{\nu_{VQ}}$ vector quantizer states per accumulator state, i.e., in total $2^{\nu_{ACC}+\nu_{VQ}}$ states. One *state transition* is labeled by the three *input* bits to the accumulator $(u_1, u_2, u_3)_{ACC}$, the virtual input bit to the VQ u_{VQ}, and by the two 4-PAM *output* symbols m_1, m_2. The intermediate outputs are the hypothesized coded bits of the accumulator $(c_1, c_2, c_3)_{ACC}$, the output of the rate $1/2$ upsampler $c_{up,1}, c_{up,2}$, the hypothesized coded bits of the VQ $c_{VQ,1}, c_{VQ,2}$, and the sign–bits $c_{s,1} = c_{up,1} + c_{VQ,1}|_{\mathrm{mod}\,2}, c_{s,2} = c_{up,2} + c_{VQ,2}|_{\mathrm{mod}\,2}$. Thus, with inputs $(u_1, u_2, u_3)_{ACC}$ and u_{VQ}, there are $2^4 = 16$ state transitions entering and leaving each state of the trellis. *A priori* information is provided by the outer variable node decoder on the inner information bits with respect to the accumulator, i.e., on $(u_1, u_2, u_3)_{ACC}$. Note that no *a priori* information is provided on the information bits u_{VQ} of the vector quantizer. The bits u_{VQ} are "virtual": By keeping u_{VQ} undetermined ("floating"), all VQ–codewords are allowed. Of course, since the TX–VQ has taken the liberty to change the sign–bits to its liking (according to its codebook), namely, to find/shape the minimum energy sequence, all VQ–codewords are equally likely and have to be "overlayed" in the trellis structure to perform appropriate detection of the sign–bits and unsigned bits, respectively. This corresponds to the summation over the coset specified in (26).

Systematic doping can be applied at the accumulator, i.e., some of the coded bits c_{ACC} are *substituted* by the corresponding systematic bits u_{ACC}. In this particular case, we only allow systematic doping of coded bits $c_{ACC,2}, c_{ACC,3}$.

C. Soft output vector quantization

Let $\mathbf{u}_{ACC}$ denote the vector of accumulator information bits (length N), and $\mathbf{u}_{VQ}$ the vector of all $N/3$ information bits of the vector quantizer. The APP vector quantizer and accumulator decoder computes the *a posteriori* L–values on the accumulator information bits u_{ACC} as

$$L_D\left(u_{ACC,k}\,|\mathbf{y}\right) = L_A\left(u_{ACC,k}\right) + \underbrace{\ln \frac{\sum\limits_{\substack{\mathbf{u}_{ACC}\in\mathbb{U}_{k,+1}\\ \forall \mathbf{u}_{VQ}}} p\left(\mathbf{y}\,|\mathbf{u}_{ACC},\mathbf{u}_{VQ}\right)\cdot \exp\left(\frac{1}{2}\mathbf{u}_{ACC,[k]}^T\cdot\mathbf{L}_{A,[k]}\right)}{\sum\limits_{\substack{\mathbf{u}_{ACC}\in\mathbb{U}_{k,-1}\\ \forall \mathbf{u}_{VQ}}} p\left(\mathbf{y}\,|\mathbf{u}_{ACC},\mathbf{u}_{VQ}\right)\cdot \exp\left(\frac{1}{2}\mathbf{u}_{ACC,[k]}^T\cdot\mathbf{L}_{A,[k]}\right)}}_{L_E(u_{ACC,k}|\mathbf{y})} \tag{37}$$

where $\mathbf{u}_{ACC,[k]}$ denotes the sub-vector of $\mathbf{u}_{ACC}$ obtained by omitting its kth element $u_{ACC,k}$, and $\mathbf{L}_{A,[k]}$ denotes the vector of all L_A-values, also omitting $u_{ACC,k}$. Thus, L_D can be written as a sum of *a priori* L-values L_A and extrinsic L-values L_E (see, e.g., [25]). The set $\mathbb{U}_{k,+1}$ contains all 2^{N-1} bit vectors $\mathbf{u}_{ACC}$ having $u_{ACC,k} = +1$, i.e., $\mathbb{U}_{k,+1} = \{\mathbf{u}_{ACC}\,|\,u_{ACC,k} = +1\}$, and $\mathbb{U}_{k,-1} = \{\mathbf{u}_{ACC}\,|\,u_{ACC,k} = -1\}$. The *a priori* L–values are defined as

$$L_A(u_{ACC,k}) = \ln\frac{P[u_{ACC,k}=1]}{P[u_{ACC,k}=-1]}. \tag{38}$$

The evaluation of (37) is efficiently done by exploiting the underlying trellis structure [28]. For numerical stability, all computations are advantageously performed in the log–domain. The simplified log–likelihood function per trellis state transition is based on the one–dimensional modulo–metric

$$\ln p\left(y_1', y_2'\,|m_1, m_2\right) \approx \ln\sum_{k=-3}^{3}\exp\left[-\frac{\left(y_1'-m_1+4k\right)^2}{2\sigma_{VQ}^2}\right] + \ln\sum_{k=-3}^{3}\exp\left[-\frac{\left(y_2'-m_2+4k\right)^2}{2\sigma_{VQ}^2}\right] \tag{39}$$

with effective noise power at the RX–VQ input of $\sigma_{VQ}^2 = (1-\alpha)^2 P_X + \alpha^2 P_N$. The transmit power P_X is measured (per real dimension) at the TX–VQ output (histogram exhibits a truncated Gaussian–like shape in the interval $(-2, 2]$). The received symbols corresponding to this trellis state transition, after one–dimensional modulo, are denoted as y_1', y_2', with $y' = [\alpha y + u] \bmod 4\mathbb{Z}$. Note that m_1, m_2 are dependent on the hypothesized information bits of accumulator and vector quantizer, $u_{ACC,1}, u_{ACC,2}, u_{ACC,3}, u_{VQ}$, and the current state in the trellis.

VI. Extrinsic Information Transfer Curves

A. Vector quantizer with BPSK per dimension

We start with determining the *mutual information limit* of a vector quantizer with BPSK per dimension. All symbols m_1, m_2 of Sections V-B,V-C are sign–bits, and no unsigned bits are transmitted. With the chain–rule of mutual information [32], [33], [34], [35] we can compute the mutual information of an *equivalent bit channel*, i.e., the channel that, effectively, is experienced by the channel decoder after VQ APP detection. For

this, we compute the mutual information transfer curve of the VQ APP detector using *a priori* knowledge that is modeled as stemming from a binary erasure channel (BEC). We disregard the accumulator decoder for the time being, and focus on the VQ transfer curves.

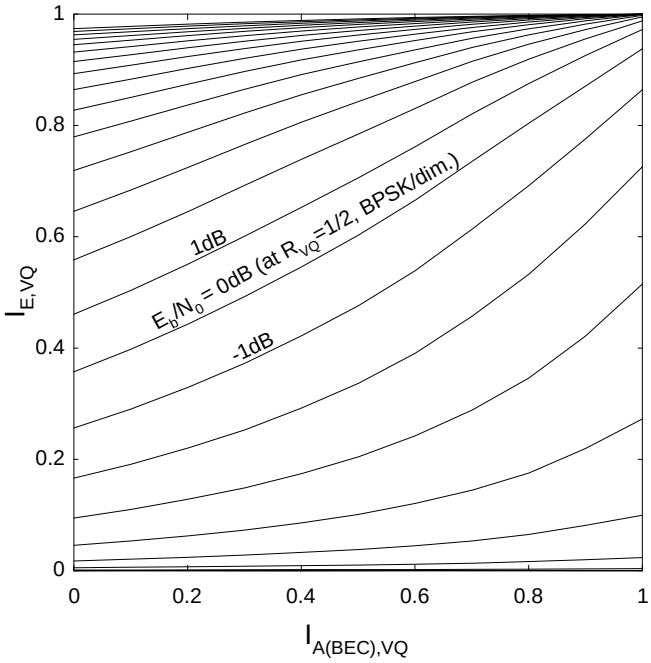

Fig. 11. Transfer curves of rate $R_{VQ} = 1/2$ vector quantizer (memory 2) and BPSK/dimension–metric for $\alpha = 0.5$ over a set of E_b/N_0–values in steps of 1dB (code rate $R_{ch} = 1$ assumed); BEC *a priori* knowledge.

Examples of such curves are depicted in Fig. 11 for a VQ of memory 2 and different E_b/N_0–values. An integration over the area under these curves yields

$$I\left(C_{ACC};\mathbf{Y}'\right) \approx \int_0^1 I_{E,VQ}\, dI_{A,VQ} \tag{40}$$

which is an estimate of the mutual information conveyed per TX–VQ (and upsampler) input bit, i.e., accumulator output bit c_{ACC}. The variable $\mathbf{Y}'$ denotes the input vector to the RX–VQ. The mutual information $I\left(C_{ACC};\mathbf{Y}'\right)$ is available to the channel decoder provided that perfect iterative decoding over detector and decoder could be performed. Hence, the EXIT transfer curves provide a convenient means for determining the mutual information limits (by numeric intergration) of the modulation in our system, i.e. the quantization encoder (Viterbi) and respective quantization detector/decoder (BCJR), independent of the specific channel code we may choose to incorporate.

By computing transfer curves over different α– and E_b/N_0–values, and numerically evaluating the corresponding area, we obtain the mutual information limits given in Fig. 12, plotted in the spectral efficiency chart. For example, for $\alpha = 0.5$ and VQ memory 2, we can use the area results $I(C_{ACC};\mathbf{Y}')$ from Fig. 11 to compute the respective mutual information limit curve in parametric form as

$$\left(10\log_{10}\left(\frac{E_b/N_0}{I\left(C_{ACC};\mathbf{Y}'\right)}\right), 2\left(1-R_{VQ}\right)\cdot I\left(C_{ACC};\mathbf{Y}'\right)\right).$$

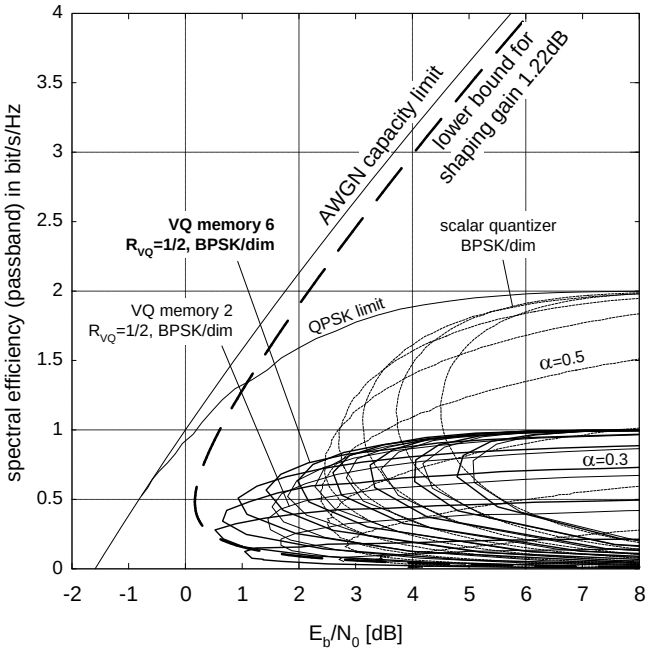

Fig. 12. Mutual information limits of VQ and BPSK/dimension for different α–values (in steps of 0.1) in the spectral effi ciency chart; $R_{VQ} = 1/2$, VQ of memory 2 and memory 6.

Note that here E_b is with respect to the accumulator output bit c_{ACC}, such that R_{ch} does not show up in the equation (equivalently, one might think of R_{ch} being set to one).

As can be seen, the advantage of a memory 6 VQ over a memory 2 VQ is bigger for smaller spectral efficiencies. As a reference, the lower bound (20) is plotted for a shaping gain of 1.22dB (memory 6 VQ). Since it is a lower bound, the actual mutual information can be better. Obviously, the "rate loss" of the rate $1/2$ VQ using BPSK per dimension leaves a gap to the AWGN capacity. In the next section we shall see how increasing the modulation from BPSK/dimension to 4-PAM/dimension helps to reduce this gap.

B. Vector quantizer with 4-PAM per dimension

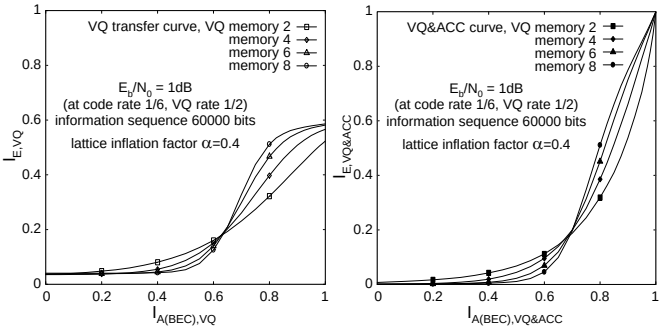

Fig. 13. Left: Transfer curves of rate $R_{VQ} = 1/2$ vector quantizer and 4-PAM/dimension–metric for $\alpha = 0.4$ at $E_b/N_0 = 1$dB (code rate $R_{ch} = 1/6$ assumed); BEC *a priori* knowledge. Right: Curves from left, combined with a memory one accumulator decoder.

Fig. 13 (left) shows VQ transfer curves for different memory using a 4-PAM/dimension–metric, that is, m_1, m_2 are drawn

from a 4-PAM constellation (natural labeling), with a sign–bit (most significant bit) and an unsigned bit (least significant bit). The S–shape of the curves is more pronounced for bigger memory. The area under the VQ curves increases with greater VQ memory as the shaping gains improve.

When we include the accumulator into the inner detector, and use the joint trellis processing as discussed in Section V, we obtain the curves depicted in Fig. 13 (right), which, now, go up to $(1,1)$, an essential requirement for good performance of iterative decoding. The accumulator is information preserving, and thus, for the same VQ memory, we get the satisfying result that the area measurements under the curves (from left chart to right chart) remain unchanged, $\int_0^1 I_{E,VQ}\, dI_{A,VQ} = \int_0^1 I_{E,VQ\&ACC}\, dI_{A,VQ\&ACC}$.

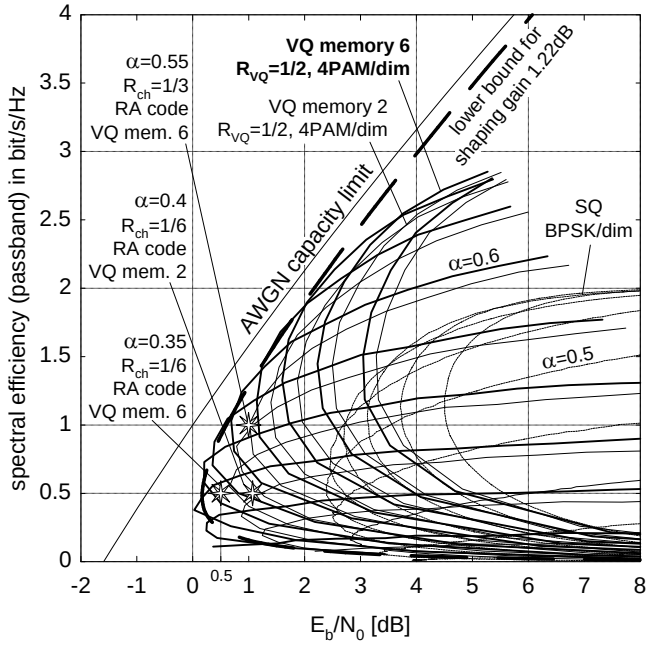

Fig. 14. Mutual information limits of VQ and 4-PAM/dimension for different α–values (in steps of 0.1); $R_{VQ} = 1/2$, VQ of memory 2 and memory 6; RA code turbo cliff position at 1.1dB (memory 2 VQ), 0.5dB (memory 6 VQ) for 0.5bit/s/Hz, and at 1dB (memory 6 VQ) for 1bit/s/Hz (code design, see Section VII).

As for the BPSK–case, we compute VQ transfer curves over different α– and E_b/N_0–values, and obtain the respective mutual information limits as depicted in Fig. 14 by plotting

$$\left(10\log_{10}\left(\frac{E_b/N_0}{I\left(C_{ACC};\mathbf{Y}'\right)}\right), 2\left(2-R_{VQ}\right)\cdot I\left(C_{ACC};\mathbf{Y}'\right)\right).$$

Again, $I(C_{ACC};\mathbf{Y}')$ is the mutual information conveyed per accumulator output bit c_{ACC}, gained through area integration; and again, in the equation, E_b is with respect to the accumulator output bit c_{ACC}, such that R_{ch} is irrelevant.

Obviously, the 4-PAM–metric allows a potential dirty paper coding scheme to approach the AWGN capacity limit much closer. Moreover, the lower bound (20) indicates that it is not required to further increase the modulation alphabet for spectral efficiencies below 2bit/s/Hz. Note that the mutual information limits are obtained using area integration over EXIT curves. We still need to design an appropriate *iterative* decoding scheme to materialize these gains. How to construct repeat–accumulate codes with the performance given by the three data points in Fig. 14 is discussed in the subsequent sections.

VII. Code Design Examples

In the previous section we focused on area integration over extrinsic information transfer curves to obtain mutual information limits plotted in the spectral efficiency chart. The shape of the transfer curves was not of particular interest. In the following we design repeat–accumulate codes using curve fitting, and the shape of the inner VQ&ACC&CND–transfer curves becomes of prime importance.

We designed RA codes of rate $R_{ch} = 1/3$ and $1/6$, to obtain an overall spectral efficiency of 1bit/s/Hz and 0.5bit/s/Hz, respectively. The EXIT chart technique was used to find appropriate VND degree distributions. For this, the outer VND transfer curve is matched to the inner VQ&ACC&CND–curve by means of curve fitting (for details see [30], [31]). In the following, we briefly review EXIT chart based code design, show how to apply systematic doping to facilitate the use of higher memory vector quantizers, and illustrate the usefulness of the proposed techniques by three code design examples. All designed codes are verified by bit error rate (BER) simulations.

A. EXIT curve of outer VND code mixtures

For a variable node of degree d_v the decoder output is $L_{i,out} = \sum_{j\neq i} L_{j,in}$, where $L_{j,in}$ is the jth *a priori* L–value going into the variable node, and $L_{i,out}$ is the ith extrinsic L–value coming out of the variable node. The $L_{j,in}$ are modeled as the output L–value of an AWGN channel whose input was the jth interleaver bit transmitted using BPSK. The EXIT function of a degree–d_v variable node is then [30]

$$I_{E,VND}(I_{A,VND}, d_v) = J\left(\sqrt{(d_v-1)}\cdot J^{-1}(I_{A,VND})\right) \tag{41}$$

with

$$J(\sigma) = 1 - \int_{-\infty}^{\infty} \frac{e^{-\left(\xi-\sigma^2/2\right)^2/2\sigma^2}}{\sqrt{2\pi}\sigma}\cdot\log_2\left[1+e^{-\xi}\right]\, d\xi. \tag{42}$$

Some of these curves are plotted in Fig. 15 for different variable node degrees.

Let D_v be the number of different variable node degrees, and denote these by $\widetilde{d}_{v,i}$, $i = 1,\ldots,D_v$. The average variable node degree is $\overline{d}_v = \sum_{i=1}^{D_v} a_{v,i}\cdot\widetilde{d}_{v,i}$, where $a_{v,i}$ is the fraction of variable nodes having degree $\widetilde{d}_{v,i}$. Let $b_{v,i}$ be the fraction of *edges* incident to variable nodes having degree $\widetilde{d}_{v,i}$. The EXIT curve of a mixture of codes is an average of the component EXIT curves [30], [34], and thus the VND curve writes as

$$I_{E,VND}\left(I_{A,VND}\right) = \sum_{i=1}^{D_v} b_{v,i}\cdot I_{E,VND}\left(I_{A,VND},\widetilde{d}_{v,i}\right). \tag{43}$$

Only $D_v - 2$ of the $\widetilde{d}_{v,i}$ can be adjusted freely because we must enforce $\sum_i b_{v,i} = 1$ and $R_{ch} = \overline{d}_c/\overline{d}_v$, with $\overline{d}_c$ being the average check node degree. Thus, we must choose $D_v \geq 3$ to permit curve fitting.

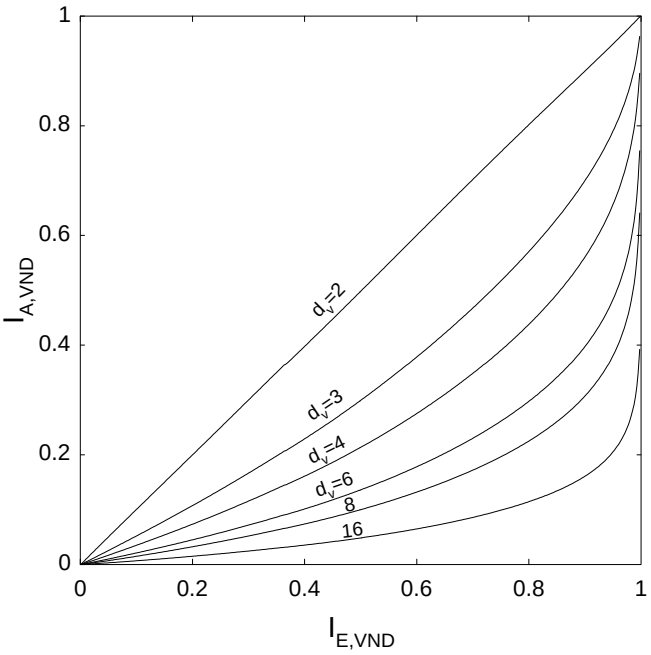

Fig. 15. VND EXIT curves for nonsystematic RA codes (axes swapped).

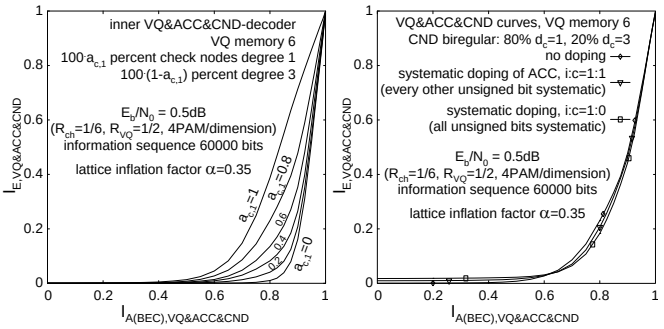

Fig. 16. Left: Transfer curves of VQ&ACC&CND with VQ of memory 6 and different check node mixtures $a_{c,1}$ of degree 1 and degree 3. Right: Transfer curves with $a_{c,1} = 0.8$ and different systematic doping ratios, to enable convergence of iterative decoding.

B. EXIT curve of inner decoder with systematic doping

We design an RA code of rate $R_{ch} = 1/6$ using a VQ of memory 6 and $R_{VQ} = 1/2$, to get a spectral efficiency of 0.5bit/s/Hz. ¿From Fig. 14 we observe that a lattice inflation factor of $\alpha \approx 0.35$ allows the closest approach to the AWGN capacity. Leaving a design margin of about 0.4dB to the mutual information limit, we use an E_b/N_0–value of 0.5dB for curve fitting, performed in the next section. First, we need to study the different shapes of the inner transfer curves which shall become our target functions for curve fitting.

The resulting EXIT curve of a mixture of VND curves with different degrees is concave. Thus, to simplify the matching procedure we need to provide an inner transfer curve which is smooth and without turning points. Fig. 16 (left) shows transfer curves for a memory 6 VQ with a biregular check node decoder: A fraction $a_{c,1}$ of the check nodes has degree $d_c = 1$, and a fraction $1 - a_{c,1}$ of the check nodes has degree $d_c = 3$. The average check node degree is then $\bar{d}_c = a_{c,1} \cdot 1 + (1 - a_{c,1}) \cdot 3$. In the figure, $a_{c,1}$ is varied from $a_{c,1} = 0$ (all check node degrees are $d_c = 3$) to $a_{c,1} = 1$ (all check node degrees are $d_c = 1$), in steps of 0.2. Obviously, the turning point (S-shape) can be smoothened out by using a moderate fraction of degree 3 check nodes, e.g., $a_{c,1} = 0.8$. However, even with all check nodes set to 1, the VQ&ACC&CND–curve starts virtually at the origin, thus preventing an iterative decoding scheme from starting to converge. We already observed this behavior in Fig. 13 (right) for VQ memories greater than two.

A simple yet effective means for solving this problem is to apply *systematic doping*. Feeding through some uncoded systematic (information) bits, i.e., bypassing the accumulator of the RA code, shifts up the inner transfer curve at the beginning, at the cost of losing some extrinsic output for higher *a priori* input (Fig. 16, right). We only dope those bits of the accumulator that are mapped onto unsigned bits of the 4-PAM constellations. A doping ratio of systematic bits to coded bits of $i : c = 1 : 1$ (such that every other unsigned bit is a systematic bit with respect to the accumulator) turned out to be sufficient to trigger convergence in the case of a memory 6 VQ.

It is interesting to note that we now use *two* forms of doping: 1.) A *biregular* CND (i.e., a fraction of the check nodes has degree $d_c = 1$) ensures that the inner ACC&CND-curve starts at a value $I_{E,ACC\&CND} > 0$, and thus allows to use a nonsystematic RA code. However, when combining the $ACC\&CND$ with a VQ of memory 6, the biregularity is not sufficient to enable convergence. 2.) In addition to that, we need to apply *systematic doping* to the ACC, and by this, in fact, making the RA code partially systematic again.

C. Code designs and simulation results

We chose vector quantizers of rate $R_{VQ} = 1/2$, memory 2 and memory 6, with feedforward polynomials $(07_8, 05_8)$ and $(0133_8, 0171_8)$, respectively. A 4-PAM constellation was applied per dimension using natural labeling. The inner detector curve (including VQ&4-PAM, ACC and CND) was computed by Monte–Carlo simulation, assuming a Gaussian model for the *a priori* information. We designed two rate $R_{ch} = 1/6$ RA codes, and one $R_{ch} = 1/3$ RA code. The codeword length is $K = 6 \cdot 10^4$ information bits, $N = 3.6 \cdot 10^5$ coded bits for the rate $1/6$ code, and $N = 1.8 \cdot 10^5$ for the rate $1/3$ code, respectively. The check node layer is biregular, with 80% of the check nodes being degree 1, and 20% being degree 3 for the rate $1/6$ code. For the rate $1/3$ code, 20% of the check nodes are degree 1, and 80% degree 3,

For the system with VQ of memory 2 ($R_{ch} = 1/6$), curve fitting at $E_b/N_0 = 1$dB yields a VND degree distribution of 64.36% variable nodes being degree 3, 31.24% degree 10, and 4.402% degree 76. We achieve convergence at 1.1dB ($\alpha = 0.4$) and plot this point in Fig. 14. No error floor was observed for 40 blocks simulated, which can be attributed to the fact that there are no degree 2 variable nodes, and the lowest variable node degree is 3. The iterations required varied from 60 to 90. Fig. 17 shows inner and outer transfer curves, and a simulated decoding trajectory at 1.2dB. The trajectory follows the individual transfer curves reasonably well.

For the VQ of memory 6 ($R_{ch} = 1/6$), curve fitting at

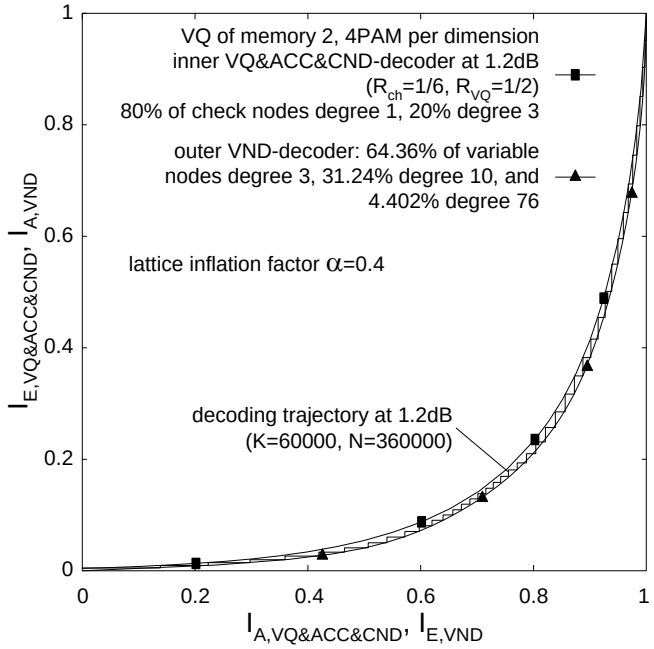

Fig. 17. EXIT chart of rate $R_{ch} = 1/6$ nonsystematic RA codes designed by curve fitting, with inner VQ&4-PAM, ACC, CND–curve and outer VND–curve; codeword length $N = 3.6 \cdot 10^5$ bits. $E_b/N_0 = 1.2$dB, VQ memory 2, no inner systematic doping.

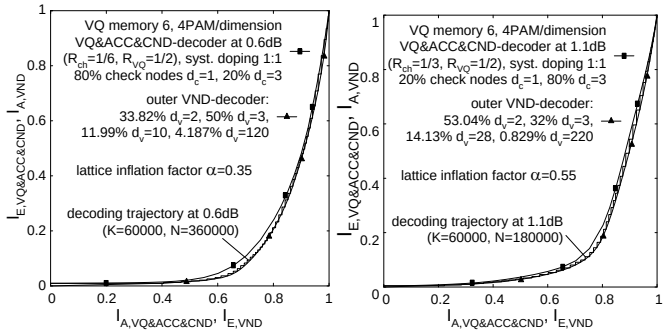

Fig. 18. EXIT charts of nonsystematic RA codes at $E_b/N_0 = 0.6$dB and 1.1dB; codeword length $N = 3.6 \cdot 10^5$ bits ($R_{ch} = 1/6$, left) and $1.8 \cdot 10^5$ bits ($R_{ch} = 1/3$, right); VQ memory 6, with inner systematic doping.

$E_b/N_0 = 0.5$dB yields a VND degree distribution of 33.82% variable nodes being degree 2, 50% degree 3, 11.99% degree 10, and 4.187% degree 120. We achieve convergence at 0.5dB (only 1.3dB away from the AWGN capacity limit at 0.5bit/s/Hz), with $\alpha = 0.35$. After 10 blocks simulated, the error floor was $3 \cdot 10^{-6}$. The iterations required varied from 75 to 115. Note that the accumulator was doped, with $i : c = 1 : 1$, i.e., every other unsigned bit was systematic with respect to the accumulator. Likewise, for the VQ of memory 6 and $R_{ch} = 1/3$, curve fitting at $E_b/N_0 = 1.0$dB yields a VND degree distribution of 53.0422% variable nodes being degree 2, 32.0% degree 3, 14.1291% degree 28, and 0.8286% degree 220. We achieve convergence at 1.0dB (only 1.0dB away from the AWGN capacity limit at 1bit/s/Hz), with $\alpha = 0.55$. After 10 blocks simulated, the error floor was $8 \cdot 10^{-6}$ (65 to 90 iterations required). As before, the accumulator was doped, with $i : c = 1 : 1$. For both codes, the corresponding simulated decoding trajectories are shown in Fig. 18 at 0.6dB and 1.1dB, respectively. Apparently, there is a mismatch between predicted and actual behavior of the inner APP processing block. The inner transfer curve was computed assuming a Gaussian model for the *a priori* knowledge; as it turns out, the predicted inner extrinsic output is too optimistic for medium I_A–values. A closer look at the histograms shows that the distributions have a significant portion of reliability values clustered around zero (erasures), owing to the sign–bits which become available rather late (opposed to the unsigned bits), at high I_A–values. This effect is stronger the higher the memory of the VQ. Thus, the Gaussian assumption is a poor model in this case, and a mixed Gaussian/erasure model would be more appropriate. However, by taking into account the overly optimistic behavior of the inner transfer curve for medium I_A–values in the curve fitting, good VND–distributions can still be found.

VIII. Further Research

The results presented were achieved using iterative decoding over an inner vector quantizer and an outer channel code. The vector quantizer itself was conventional, based on a simple convolutional code as the "quantizer code". A gap of about 1.3dB remains to the AWGN capacity at 0.5bit/s/Hz spectral efficiency when using a vector quantizer of memory 6. The gap widens up to 2.1dB at zero spectral efficiency, a regime that may be important in digital watermarking, see [36]. To further increase the shaping gain, we need to increase the memory of the convolutional code, and move from a binary to a multilevel quantizer code. However, the complexity grows exponentially with the code memory, and the system presented (memory 6) is already quite complex. In channel coding, the discovery of turbo codes and iterative decoding over simple component codes avoided the exponential complexity growth while improving the coding gain. Unfortunately, thus far, there has not been similar progress in quantization, where convolutional codes coupled with the Viterbi algorithm still offer the best performance. An effective technique to achieve more powerful shaping has yet to be found.

IX. Summary

We presented a realization of a multidimensional dirty paper coding scheme that offers substantial gains over one–dimensional scalar quantization. While for scalar quantization a simple AWGN turbo code together with a modulo metric is sufficient to achieve reliable communication close to its mutual information limits, the gap to capacity of such systems is large at low SNR. For vector quantization we showed how to perform iterative quantization and decoding using a nonsystematic repeat–accumulate code. The design was exemplified for systems operating at low spectral efficiency. The improvement is more than 2dB over the best scalar quantizer.

Acknowledgments

The authors would like to thank J. Salz and G. J. Foschini for providing the initial impetus for the endeavor reported in this work.

REFERENCES

[1] U. Erez, S. Shamai (Shitz), and R. Zamir, "Capacity and lattice strategies for cancelling known interference," *IEEE Trans. Information Theory*, submitted June 2003, see also *Proc. ISITA, Honolulu, Hawaii*, pp. 681–684, Nov. 2000.

[2] M. H. M. Costa, "Writing on dirty paper," *IEEE Trans. Inf. Theory*, IT-29, pp. 439–441, May 1983.

[3] S. I. Gelfand and M. S. Pinsker, "Coding for channel with random parameters," *Problemy Pered. Inform. (Problems of Inform. Trans.)*, vol. 9, no. 1, pp. 19–31, 1980.

[4] F. M. J. Willems, "On Gaussian channels with side information at the transmitter," *Proc. of the Nineth Symposium on Inf. Theory in the Benelux*, Enschede, The Netherlands, 1988.

[5] P. Moulin and J. O'Sullivan, "Information-theoretic analysis of information hiding," 1999.

[6] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *Proc. Int. Symp. Inform. Theory (ISIT)*, p. 46, June 2000.

[7] J. J. Eggers, J. K. Su, and B. Girod, "A blind watermarking scheme based on structured codebooks," *IEE Colloquium: Secure images and image authentication, London, UK*, Apr. 2000.

[8] G. Caire and S. Shamai (Shitz), "On the achievable throughput of a multiantenna Gaussian broadcast channel," *IEEE Trans. Inf. Theory*, submitted. See also, "On achievable rates in multiple-antenna broadcast downlink", *38th Allerton Conf. on Commun., Control and Computing*, Oct. 2000, and the *Proceedings of the 2001 IEEE International Symposium on Information Theory*, Washington, D.C., U.S.A., p. 147, June 2001.

[9] W. Yu and J. Cioffi , "Sum capacity of a Gaussian vector broadcast channel," *IEEE Trans. Inf. Theory*, Nov. 2001.

[10] S. Viswanath, N. Jindal, and A. Goldsmith, "On the capacity of multiple input multiple output broadcast channels," *Proc. Int. Conf. on Commun. (ICC)*, May 2002.

[11] P. Viswanath and D. Tse, "Sum capacity of the multiple antenna broadcast channel," *Int. Symp. Inf. Theory*, July 2004.

[12] T. Philosof, U. Erez, and R. Zamir, "Combined shaping and precoding for interference cancellation at low SNR," *Proc. of Int. Symp. Inform. Theory*, p. 68, June 2003.

[13] G. D. Forney, Jr., "Trellis shaping," *IEEE Trans. Inf. Theory*, IT-38, pp. 281–300, Mar. 1992.

[14] M. V. Eyuboglu and G. D. Forney, Jr., "Trellis precoding: Combined coding, precoding and shaping for intersymbol interference channels," *IEEE Trans. Inf. Theory*, IT-38, pp. 301–314, Mar. 1992.

[15] R. F. Fischer, C. Windpassinger and J. B. Huber, "Modulo-Lattice Reduction in Precoding Schemes," *Proc. of Int. Symp. Inform. Theory (ISIT2003), Yokohama, Japan*, p. 163, June 2003.

[16] R. Zamir, S. Shamai (Shitz), and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inf. Theory*, IT–48, pp. 1250 –1276.

[17] G. D. Forney, Jr., "On the role of MMSE estimation in approaching the information-theoretic limits of linear Gaussian channels: Shannon meets Wiener," *41st Allerton Conf. on Commun., Control, and Computing*, Oct. 2003.

[18] G. D. Forney, Jr., "Coset codes-I: Introduction and geometrical classification," *IEEE Trans. Inf. Theory*, IT–34, pp. 1123 –1151, Sept. 1988.

[19] R. Zamir and M. Feder, "On lattice quantization noise," *IEEE Trans. Information Theory*, IT-42, pp. 1152–1159, July 1996.

[20] M. Tomlinson, "New automatic equalizer employing modulo arithmetic," *Electronic Lett.*, vol. 7, pp. 138–139, Mar. 1971.

[21] H. Harashima and H. Miyakawa, "Matched-Transmision Technique for Channels with Intersymbol Interference," *IEEE Trans. Commun.*, COM–20, pp. 774–780, Aug. 1972.

[22] A. R. Calderbank, L. H. Ozarow, "Nonequiprobable Signaling on the Gaussian Channel," *IEEE Trans. Inf. Theory*, vol. 36, no. 4, July 1990.

[23] G. Caire and S. Shamai, "Writing on dirty tape with LDPC codes," *DIMACS Workshop on Signal Processing for Wireless Transmission, Rutgers University, NJ, USA*, Oct. 7-9, 2002.

[24] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error–correcting coding and decoding: Turbo–codes," *Proc. Int. Conf. on Commun. (ICC)*, pp. 1064–1070, May 1993.

[25] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, IT-42, no. 2, pp. 429–445, Mar. 1996.

[26] J. H. Conway, N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. New York: Springer–Verlag, 1988.

[27] G. D. Forney, Jr., "Multidimensional constellations — Part II: Voronoi constellations," *IEEE Journ. on Select. Areas in Commun.*, vol. 7, no. 6, pp. 941–958, Aug. 1989.

[28] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Information Theory*, IT-20, pp. 284–287, Mar. 1974.

[29] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat–accumulate codes," *Proc. 2nd Int. Symp. Turbo Codes and Related Topics*, pp. 1–8, Sept. 2000.

[30] S. ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Trans. Sign. Proc. (Special Issue on MIMO Proc. Techn.)*, vol. 51, no. 11, pp. 2764-2772, Nov. 2003.

[31] S. ten Brink and G. Kramer, "Turbo processing for scalar and vector channels," *Proc. 3rd Int. Symp. Turbo Codes and Related Topics*, pp. 23–30, Sep. 2003.

[32] S. ten Brink, "Exploiting the chain rule of mutual information for the design of iterative decoding schemes," *39th Annual Allerton Conf. on Commun., Control, and Computing*, Oct. 2001.

[33] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic Information Transfer Functions: Model and Erasure Channel Properties," *IEEE Trans. Inf. Theory*, to appear 2004.

[34] M. Tüchler, J. Hagenauer, "EXIT charts and irregular codes," *36th Ann. Conf. on Inf. Sci. and Syst.*, Princeton, Mar. 2002.

[35] H. D. Pfi ster and P. H. Siegel,"Joint iterative decoding of LDPC codes and channels with memory," *Proc. 3rd Int. Symp. Turbo Codes and Related Topics*, pp. 15–18, Sep. 2003.

[36] P. Moulin, A. K. Goteti and R. Kötter "Optimal Sparse-QIM Codes for Zero-Rate Blind Watermarking," *ICASSP*, 2004.