Near-Capacity Dirty-Paper Code Design: A Source-Channel Coding Approach

Yong Sun, *Member, IEEE*, Yang Yang, *Member, IEEE*, Angelos D. Liveris, *Member, IEEE*, Vladimir Stanković, *Member, IEEE*, and Zixiang Xiong, *Fellow, IEEE*

***Abstract*—This paper examines near-capacity dirty-paper code designs based on source–channel coding. We first point out that the performance loss in signal-to-noise ratio (SNR) in our code designs can be broken into the sum of the packing loss from channel coding and a modulo loss, which is a function of the granular loss from source coding and the target dirty-paper coding rate (or SNR). We then examine practical designs by combining trellis-coded quantization (TCQ) with both systematic and nonsystematic irregular repeat–accumulate (IRA) codes. Like previous approaches, we exploit the extrinsic information transfer (EXIT) chart technique for capacity-approaching IRA code design; but unlike previous approaches, we emphasize the role of strong source coding to achieve as much granular gain as possible using TCQ. Instead of systematic doping, we employ two relatively shifted TCQ codebooks, where the shift is optimized (via tuning the EXIT charts) to facilitate the IRA code design. Our designs synergistically combine TCQ with IRA codes so that they work together as well as they do individually. By bringing together TCQ (the best quantizer from the source coding community) and EXIT chart-based IRA code designs (the best from the channel coding community), we are able to approach the theoretical limit of dirty-paper coding. For example, at 0.25 bit per symbol (b/s), our best code design (with 2048-state TCQ) performs only 0.630 dB away from the Shannon capacity.**

***Index Terms*—Dirty-paper coding, extrinsic information transfer (EXIT) chart, irregular repeat–accumulate (IRA) codes, modulo loss, packing loss, trellis-coded quantization (TCQ).**

I. Introduction

COMMUNICATING over side-information channels constitutes a powerful paradigm that is built upon the framework of channel coding with side information (CCSI), where the channel state is known only to the encoder as side information. CCSI has recently generated much research interests due to its applications to data hiding [8], [20], precoding for interference channels [11], coding for multiple-input multiple-output (MIMO) broadcast channels [32], and transmitter cooperation [17] in *ad hoc* networks. However, although the capacity results obtained by Gelfand and Pinsker [14] and by Costa on the special case of the so-called problem of "writing on dirty paper" [7] have been known for more than 20 years, practical approaches to CCSI did not appear until 1999.

The first practical dirty-paper code was developed by Chen and Wornell under the name of quantization index modulation [3]. Eggers *et al.* [9] proposed a scalar Costa scheme for information embedding. Yu *et al.* [37] employed trellis-coded quantization (TCQ) [19] as the source code and trellis-coded modulation (TCM) [29] as the channel code. These dirty-paper coding (DPC) schemes perform several decibels away from the capacity. Another implementation using turbo-coded trellis-based construction was designed by Chou *et al.* [4]. However, the employed TCQ source code is significantly weakened due to its coupling with turbo code, resulting in dimensional mismatch [25] between the source code and the channel code. Indeed, at a transmission rate of 1.0 bit per sample (b/s), the design of [4] performs 2.02 dB away from the capacity. An improved scheme based on nested turbo codes [25] performs 1.42 dB way from the capacity at 1.0 b/s.

The designs above do not perform well at low rate where the dirty-paper code design is more challenging due to magnification of the granular loss of the source code (see Section II-B). Recent works by Erez and ten Brink [10] and by Bennatan *et al.* [2] target at the low-rate regime. However, the code designs in [10], [2] still show a 1.3-dB gap from the capacity-achieving signal-to-noise ratio (SNR) at 0.25 b/s.

This paper proposes a source–channel coding approach to DPC. In addition to channel coding, strong source coding is employed to satisfy the power constraint due to the presence of encoder side information. Since the basic element of CCSI is *binning*, to justify our source–channel coding based approach, we start with an information-theoretic interpretation of algebraic binning [38]. Then, building upon results in [10], [38], we state that the performance loss (in SNR) in our practical code designs is the sum of the packing loss from channel coding and a *modulo loss*, which is a function of both the granular loss from source coding and the target rate (or SNR). At high rate, the modulo loss is approximately equal to the granular loss, but as the rate decreases, it becomes higher and higher than the granular loss. Thus, besides advanced channel codes, it is imperative

Manuscript received July 31, 2006; revised March 23, 2009. Current version published June 24, 2009. This work was supported in part by the National Science Foundation (NSF) under Grant 0729149 and Qatar National Research Fund under NPRP 28-6-7-28 and 08-101-2-025. The material in this paper was presented in part at the 39th Annual Conference on Information Science and Systems, Baltimore, MD, March 2005 and IEEE International Symposium on Information Theory, Adelaide, Australia, September 2005.

Y. Sun is with Lemko Corporation, Schaumburg, IL 60194 USA (e-mail: yong.sun@lemko.com).

Y. Yang and Z. Xiong are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: yangyang@ece.tamu.edu; zx@ece.tamu.edu).

A. Liveris is with Microwave Networks Inc., Stafford, TX 77477 USA (e-mail: alive@ieee.org).

V. Stanković is with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK G1 1XW, U.K. (e-mail: vladimir.stankovic@eee.strath.ac.uk).

Communicated by M. P. Fossorier, Associate Editor for Coding Techniques.

Color versions of Figures 2–8, 10, and 12–21 in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2009.2021319

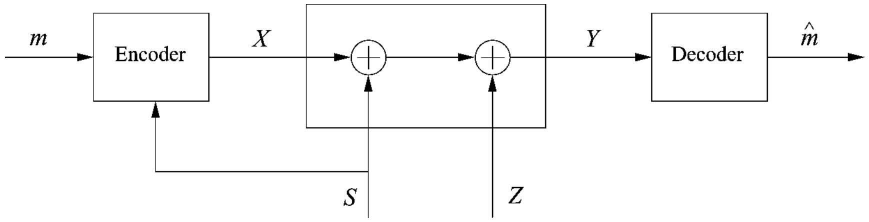

Fig. 1. Gelfand–Pinsker coding or CCSI at the encoder.

to employ strong source codes so that the granular loss is small enough for near-capacity DPC, especially at low rate.

We follow a simple design philosophy of *picking a strong source code first and then focusing on designing near-capacity channel codes*. Specifically, we present practical code designs using TCQ [19] in conjunction with both nonsystematic and systematic irregular repeat–accumulate (IRA) codes [18], and point out that the latter offers additional flexibility in allowing more design options for the systematic part. We choose TCQ because it is the best quantizer from the source coding community and endeavor to achieve as much granular gain as possible. The aim of IRA coding then is to approach the capacity of the equivalent modulo lattice channel [10], [11], [38] induced by TCQ. To synergistically combine TCQ and IRA codes together, we introduce a *tunable shift* between TCQ codebooks corresponding to different input messages, and optimize this shift for the best channel coding performance–complexity tradeoff.

Erez and ten Brink [10] used systematic doping in addition to two–dimensional (2-D) vector quantization (VQ) [15] in conjunction with trellis shaping [13] for source coding and IRA codes for channel coding. As explained in Section III, the role of the optional systematic doping step is to facilitate IRA code design, and VQ followed by trellis shaping only delivers subpar source coding performance when compared to TCQ. In our work, we directly tackle the problem of designing IRA codes without systematic doping for the TCQ-induced (modulo lattice) channel. Instead of using systematic doping in the accumulator (ACC), each output bit of the ACC is used to select TCQ codebook from two candidates that are shifted versions of each other. Clearly, shifting the codebook will not affect the quantization performance of TCQ, however, the relative shift between the two TCQ codebooks affects both the capacity of the TCQ-induced channel and the y-intercept of the ACC extrinsic information transfer (EXIT) chart. We optimize this relative shift to achieve the best tradeoff between the design complexity and coding performance of IRA codes. This way, the performance of TCQ is never compromised for easier IRA code design and the end results are dirty-paper codes with better performance at lower complexity than those in [10]. For example, at 0.25 b/s, our best code design (with 2048-state TCQ) performs only 0.630 dB away from the capacity. This improves our preliminary results published in [23], [24]. We are thus approaching the limit of DPC.

The rest of the paper is organized as follows. Section II gives background information on CCSI at the encoder, lattice precoding, and TCQ. Section III motivates our new code designs. Section IV details our dirty-paper code designs, and Sections V and VI present simulations results using TCQ and nonsystematic and systematic IRA codes, respectively. Section VII discusses computational complexity, while Section VIII concludes the paper.

II. Background

A. CCSI at the Encoder

In Gelfand–Pinsker coding [14], a transmitter wishes to communicate messages to a receiver over a noisy channel. Thus, the transmitter sends encoded messages, $m \in \{1, \dots, T\}$, over a memoryless channel defined by the transition probabilities $p(y \,|\, x, s)$, where X and Y are the channel input and output, respectively, and the independent and identically distributed (i.i.d.) random variable S is the state of the channel (side information) known non-causally to the encoder but not to the decoder. Based on the selected message t and the state of the channel S, the encoder sends codeword X which must satisfy the power constraint $E[w(X,S)] \le P_X$, with $w(\cdot,\cdot)$ being the cost measure. The achieved transmission rate is then $R = \frac{1}{N}\log_2 T$ bits per second, where N is the number of transmitted symbols.

The capacity is given by [14]

$$C^* = \max_{p(u,x\,|\,s)} [I(U;Y) - I(U;S)] \tag{1}$$

where U is an auxiliary random variable such that two Markov chains $Y \to (X,S) \to U$ and $Y \to (U,S) \to X$ hold, and $E[w(X,S)] \le P_X$. Note that the cardinality of U is no larger than the sum of cardinalities of X and S, i.e., $|\mathcal{U}| \le |\mathcal{X}| + |\mathcal{S}|$ [14]. A special case of Gelfand–Pinsker coding is shown in Fig. 1, where X and Y are related by

$$Y = X + S + Z \tag{2}$$

and Z is the noise in the channel.

Gelfand–Pinsker coding in general suffers capacity loss when compared to channel coding with side information available at both the encoder and the decoder. However, when S and Z are mutually independent i.i.d. Gaussian random variables with variances P_S and P_Z, respectively, we have the celebrated DPC problem [7] without capacity loss, i.e.,

$$C^* = \frac{1}{2}\log_2\left(1 + \frac{P_X}{P_Z}\right). \tag{3}$$

This no capacity loss result is later generalized to the case when S is arbitrarily distributed and Z is i.i.d. Gaussian and independent of S [5], [11].

Costa's proof [7], which shows the existence of capacity-achieving random binning schemes, is not constructive; thus, it does not provide any indication about practical code construc-

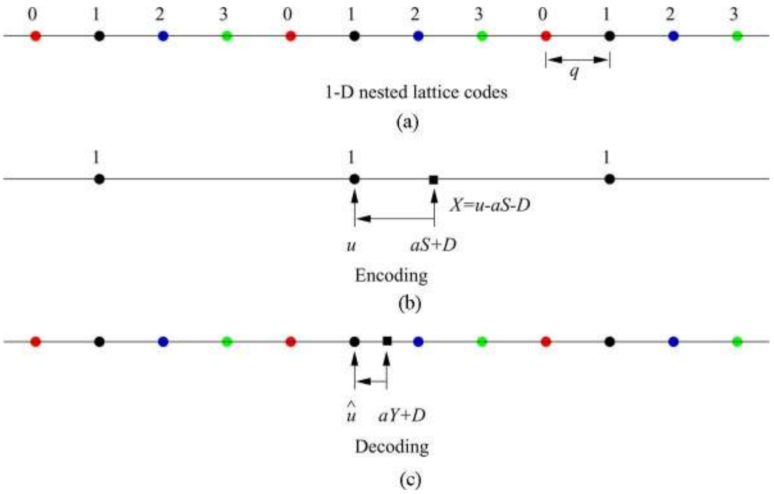

Fig. 2. 1-D nested lattice codes for DPC.

tion. Willems [33] was the first to propose quantization of the encoder side information S in the context of DPC (with causally known S). Zamir *et al.* [38] suggested an algebraic binning scheme based on nested lattice codes. The scheme consists of a coarse lattice code nested within a fine lattice code.

Fig. 2(a) illustrates one–dimensional (1-D) nested lattice/scalar codes with an infinite uniform constellation, where q denotes the quantization step size. The channel codewords are grouped into cosets/bins (labeled as 0, 1, 2, and 3) for source coding. At the encoder, the side information S is linearly scaled by α [7] and added to a uniformly distributed dither $D \in [-\frac{q}{2}, \frac{q}{2}]$ before being quantized to the closest codeword u by the source code selected by the message m to be transmitted (e.g., the coset/bin labeled 1 in Fig. 2(b)), so that the obtained quantization error $X = u - \alpha S - D$ satisfies the power constraint $E[X^2] \le P_X$. Then, X is transmitted over the additive white Gaussian noise channel with noise $Z \sim N(0, P_Z)$. According to [7], the optimal $\alpha = \frac{P_X}{P_X+P_Z} = \frac{\mathrm{SNR}}{\mathrm{SNR}+1}$, with $\mathrm{SNR} = \frac{P_X}{P_Z}$. The decoder (see Fig. 2(c)) receives the signal $Y = X + S + Z$ and finds the codeword $\hat{u}$ closest to $Y' = \alpha Y + D$. Finally, the index of the bin containing $\hat{u}$ is identified as the decoded message.

It is shown in [38] that this nested lattice scheme approaches the capacity in (3) as the dimensionality of the employed lattices approaches infinity. However, nested lattice coding typically requires the same dimensional coarse lattice source code and fine lattice channel code, which are difficult to implement in high dimensions.

B. Lattice Precoding

Let Λ be an N-dimensional lattice quantizer with basic Voronoi cell $\mathcal{V}$. Associated with $\mathcal{V}$ are several important quantities: the cell volume $|\mathcal{V}|$, the second moment $P(\Lambda)$, and the normalized second moment $G(\Lambda)$, defined by

$$|\mathcal{V}| = \int_{\mathcal{V}} d\boldsymbol{x}, \quad P(\Lambda) = \frac{1}{N|\mathcal{V}|} \int_{\mathcal{V}} |\boldsymbol{x}|^2 \, d\boldsymbol{x}, \quad \text{and} \quad G(\Lambda) = P(\Lambda)/|\mathcal{V}|^{\frac{2}{N}} \tag{4}$$

respectively. The minimum of $G(\Lambda)$ over all N-dimensional lattices is denoted as G_N. $G_N > \frac{1}{2\pi e}, \forall N$ and $\lim_{N\to\infty} G_N = \frac{1}{2\pi e}$ [6]. The granular gain of Λ is $g(\Lambda) = -10\log_{10} 12G(\Lambda)$, which is maximally 1.53 dB.

Denote S^N as a length-N source vector, and D^N as the random dither uniformly distributed over $\mathcal{V}$. Then for any source codewords (or constellation points) $u^N \in \mathcal{V}$, the encoder transmits

$$X^N = [u^N - \alpha S^N - D^N] \bmod \Lambda. \tag{5}$$

The received signal at the decoder is

$$Y^N = X^N + S^N + Z^N. \tag{6}$$

The decoder then computes

$$Y'^N = [\alpha Y^N + D^N] \bmod \Lambda = [u^N + Z'^N] \bmod \Lambda \tag{7}$$

where α is a scaling factor, and

$$Z'^N \triangleq [(1-\alpha)D^N + \alpha Z^N] \bmod \Lambda \tag{8}$$

is the equivalent modulo lattice channel noise [10], [11], [38].

The maximum achievable rate of the modulo lattice channel is $\frac{1}{N} I(U^N; Y'^N)$, achieved by a uniformly distributed input U^N over $\mathcal{V}$. Due to the dither D^N, X^N is independent of U^N and uniformly distributed over $\mathcal{V}$ with $\frac{1}{N}E[||X^N||^2] = P_X = P(\Lambda)$. Then, for $N > 1$, $\frac{1}{N}I(U^N; Y'^N)$ can be lower-bounded by assuming D^N has *i.i.d. Gaussian components* and using the mean-square error (MSE)-optimal $\alpha = \frac{P_X}{P_X+P_Z} = \frac{\mathrm{SNR}}{\mathrm{SNR}+1}$ [7], yielding [10]

$$\frac{1}{N} I(U^N; Y'^N) \ge \frac{1}{2}\log_2(1 + \mathrm{SNR}) - \frac{1}{2}\log_2 2\pi e G(\Lambda). \tag{9}$$

Note that for any finite N, the components of D^N (and X^N) are not Gaussian or independent. In practice, N has to be high for the i.i.d. Gaussian assumption to be approximately true and for the lower bound in (9) to be tight.

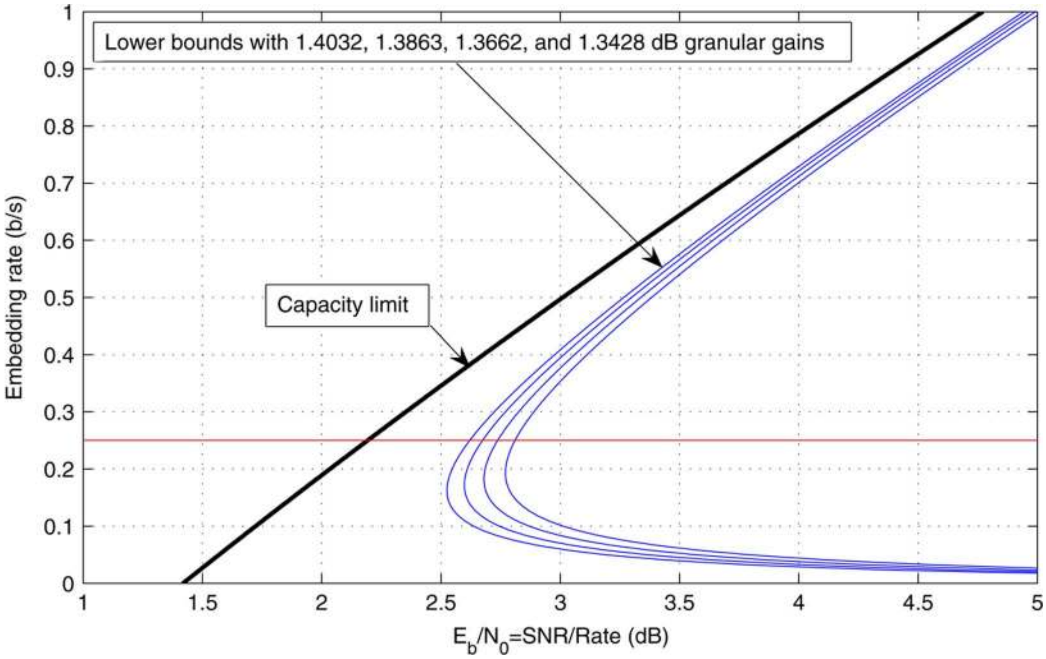

Fig. 3. The dirty-paper capacity limit and the lower bound in (9) for $g(\Lambda) = 1.4032, 1.3863, 1.3662$, and 1.3428 dB. As the rate decreases, the modulo loss $\Delta\text{SNR}_\text{m}$ in (10) increases for fixed $G(\Lambda)$.

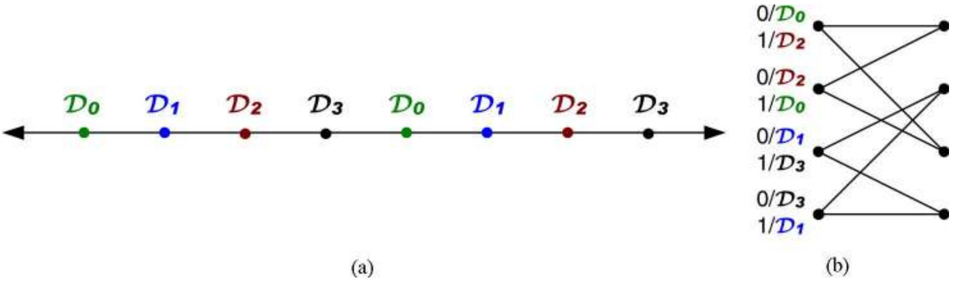

Fig. 4. TCQ codebook with subset labeling for $R = 2$ b/s and a 4-state trellis [26]. (a) Coset labeling, (b) 4-state trellis.

Expression (9) indicates that with ideal channel coding, the loss in rate due to lattice quantization (or source coding) is maximally $\frac{1}{2}\log_2 2\pi e G(\Lambda)$ bits per second. In order to measure this loss, we define the *modulo loss* (in decibels) corresponding to the lower bound in (9) due to the modulo operation in lattice quantization as

$$\Delta\text{SNR}_\text{m} \triangleq 10\log_{10}\frac{\text{SNR}}{\text{SNR}^*} = 10\log_{10}\frac{2\pi e G(\Lambda)2^{2C^*}-1}{2^{2C^*}-1} \tag{10}$$

where $\text{SNR}^* = 2^{2C^*} - 1$ is the capacity-achieving SNR. When the capacity C^* is high

$$\Delta\text{SNR}_\text{m} \approx 10\log_{10} 2\pi e G(\Lambda) = 1.53 - g(\Lambda) \text{ dB} \tag{11}$$

i.e., the modulo loss is approximately equal to the granular loss from source coding in this case. But as C^* decreases, the modulo loss $\Delta\text{SNR}_\text{m}$ increases when $G(\Lambda)$ is fixed, as indicated in Fig. 3. To reduce $\Delta\text{SNR}_\text{m}$, it is imperative to use high-dimensional lattice quantizers (or VQ in general), instead of scalar schemes such as Tomlinson–Harashima precoding [28], [16], to achieve as much granular gain as possible so that $2\pi e G(\Lambda)$ approaches one. At $C^* = 0.25$ b/s, when the granular gain $g(\Lambda)$ of Λ is 1.2698, 1.3083, 1.3428, 1.3662, 1.3863, and 1.4032 dB, $\Delta\text{SNR}_\text{m}$ equals 0.8395, 0.7234, 0.6175, 0.5447, 0.4815, and 0.4278 dB, respectively. This highlights the importance of having a strong source code in DPC, especially at low rate.

C. TCQ

TCQ [19] is the source coding counterpart of trellis-coded modulation [29] in channel coding. The basic idea of TCQ is to allow an expanded signal set and use coded modulation for set partitioning. For encoding a memoryless source using TCQ at R bits per second, a scalar codebook $\mathcal{D}$ with 2^{R+1} codewords is designed. This codebook is then partitioned into four disjoint subsets $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$, and $\mathcal{D}_4$, each with 2^{R-1} codewords. These subsets are used to label the branches of a trellis. Fig. 4 shows a TCQ codebook with subset labeling for $R = 2$ b/s and a 4-state trellis [26]. Given an input sequence, the Viterbi algorithm is used to choose the trellis path (sequence of codewords) that minimizes the MSE between the input sequence and output codewords. Each R-bit codeword consists of one bit specifying the chosen subset (trellis path) and an $(R-1)$-bit codeword necessary to specify codewords from the chosen subsets.

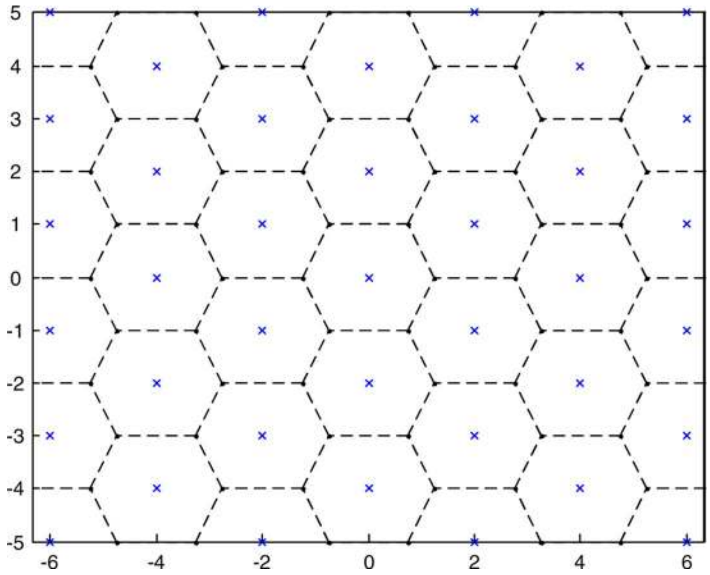

Fig. 5. Codewords and Voronoi regions of a 4-state length-2 TCQ using Fig. 4, [26].

In this paper, we consider TCQ with infinitely uniform codebook $\mathcal{D} = q\mathbb{Z}$ (i.e., $R = \infty$), and fix the step size $q = 1$ without loss of generality since the granular gain of TCQ is independent of q. Then the equivalent N-dimensional TCQ codebook can be written as

$$\Lambda = 4\mathbb{Z}^N \oplus \mathcal{C} = \{4z^N + c^N : z^N \in \mathbb{Z}^N \text{ and } c^N \in \mathcal{C}\} \quad (12)$$

where $\mathbb{Z}^N$ is the N-dimensional integer lattice and $\mathcal{C}$ is a set containing 2^N 4-ary vectors indexed by N trellis bits. For example, starting from an initial state of 0 in Fig. 4(b), a 4-state length-2 TCQ is a 2-D lattice quantizer with codebook $\Lambda = 4\mathbb{Z}^2 \oplus \mathcal{C}$ where $\mathcal{C} = \{(0,0)^T, (0,2)^T, (2,1)^T, (2,3)^T\}$. As shown in Fig. 5, the corresponding Voronoi regions are very similar to those of the optimal hexagonal lattice.

The reported granular gain in [19] of TCQ with the 256-state trellis taken from [29] is 1.33 dB. Thus, the performance of 256-state TCQ is only 0.2 dB away from the distortion–rate bound for uniform sources, which is better than any vector quantizer of dimension less than 69 [26]. With the help of entropy coding, the same 0.2-dB gap can be obtained at all rates by entropy-constrained TCQ [26]. Slightly better TCQ performances at the same number of states are obtained by searching for polynomials that maximize granular gains instead of free distances [34]. The resulting largest granular gains for 256-, 512-, 1024-, and 2048-state TCQs are 1.3428, 1.3662, 1.3863, and 1.4032 dB, respectively, which are used in Fig. 3.

III. Motivation of Our Code Designs

In [10], Erez and ten Brink propose one of the best performing dirty-paper code designs, where 2-D VQ is employed with trellis shaping after nonsystematic IRA coding and systematic doping, in which some check bits of the IRA codes are directly outputted to index the 16-QAM constellation. Systematic doping is employed to help raising the beginning part of the IRA extrinsic information transfer (EXIT) chart, so that the EXIT chart has a relatively large y-intercept compared to that without doping. With VQ plus memory-6 (or 64-state) trellis shaping and a target rate of $C^* = 0.25$ b/s, their design [10] achieves near-lossless decoding at $\mathrm{SNR} = -2.510$ dB, which is 1.32 dB away from $\mathrm{SNR}^* = -3.83$ dB. We can view the total performance loss as the sum of the packing loss from channel coding and a modulo loss. For example, the dirty-paper code [10] with VQ and memory-6 trellis shaping has a shaping gain of 1.22 dB, which corresponds to a 0.98-dB modulo loss. Hence, the total loss of 1.32 dB can be written as $1.32 = 0.98 + 0.34$ dB, where 0.34 dB is the packing loss $\Delta\mathrm{SNR}_\mathrm{p}$ due to practical IRA coding. Our implementation of the scheme in [10] using 2-D VQ and memory-8 (or 256-state) trellis shaping with 1.28-dB shaping gain gives a performance that is $0.81 + 0.34 = 1.15$ dB away from SNR^*.

Although systematic doping makes IRA code design an easier task, it is *optional* in the sense that an IRA code without doping can perform just as well. A detailed discussion on systematic doping is as follows.

- Advantage: *Increased y-intercept of check node decoder (CND) EXIT chart and easier IRA code design.* With systematic doping, a fraction of input bits to ACC are "doped" without passing through the ACC. Consequently, at the decoder, these doped bits can directly obtain information from the equivalent modulo lattice channel, while the remaining undoped bits only receive information that is degraded by the ACC. Thus, the starting value of the y-intercept of CND EXIT chart will be much higher than that without doping. Quantitatively, for Erez and ten Brink's design at $C^* = 0.25$ b/s with memory-6 trellis shaping, the y-intercept of the CND EXIT chart at $\mathrm{SNR} = -2.560$ dB is 0.01397 with systematic doping, and 0.00024 without. Since a smaller y-intercept of CND EXIT chart often requires a higher maximum variable node decoder (VND) degree, which takes more effort to design, systematic doping does help make IRA code design easier. For example, Erez and ten Brink's rate-$\frac{1}{4}$ design with memory-6 trellis shaping has a maximum VND degree of 120, but if

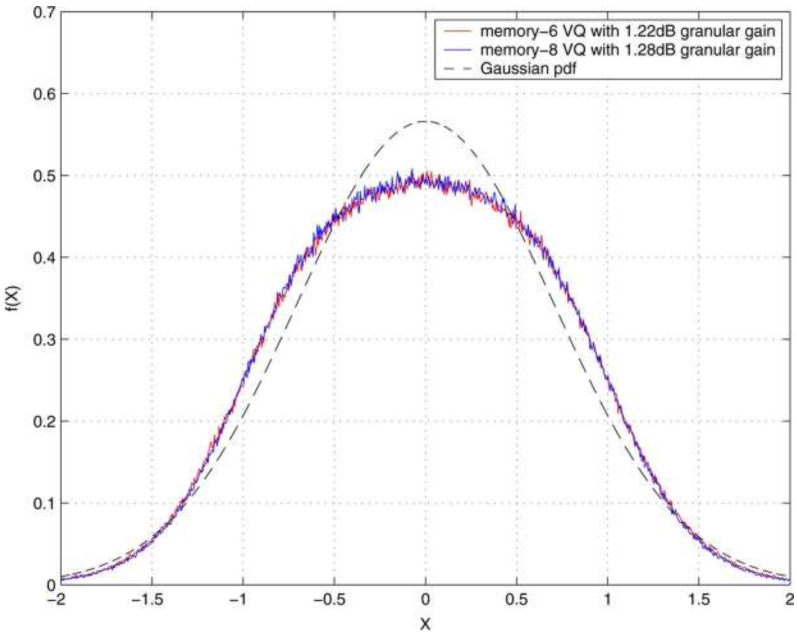

Fig. 6. The marginal distribution of X^N with 2-D VQ plus memory-6 and memory-8 shaping trellises in [10], together with a Gaussian pdf of the same variance.

systematic doping is removed, a maximum VND degree of 1747 has to be used in our implementation to ensure proper start of decoding. This advantage of systematic doping would be more pronounced at short block length, where a large VND degree might be infeasible.

- Disadvantage: *Compromised channel code performance at long block length.* Since the ACC is used to improve the distance spectrum of the resulting IRA codes, from a channel coding perspective, introducing systematic doping to ACC compromises its distance-boosting property, and hence leads to suboptimal IRA code performance. Indeed, our implementation of the scheme in [10] using 2-D VQ and memory-6 trellis shaping *without* systematic doping achieves convergence at $\mathrm{SNR} = -2.560$ dB, which is 0.05 dB better than the performance reported in [10] with doping at the same block length of 60 000 information bits. The corresponding variable node profile is

$$\lambda(x) = \sum_{d=2}^{d_v} \lambda_d x^d = 0.4825x^2 + 0.3388x^3 + 0.1481x^{11} + 0.0275x^{15} + 0.0001x^{210} + 0.0030x^{1747}$$

and the check node profile is

$$\rho(x) = \sum_{d=1}^{d_c} \rho_d x^d = 0.7222x + 0.2778x^3$$

where λ_d/ρ_d is the fraction of variable/check nodes with degree d, and d_v/d_c is the maximum variable/check node degree.

Erez and ten Brink's design [10] employs 2-D VQ in conjunction with trellis shaping for source coding. Fig. 6 depicts the marginal probability density function (pdf) of X^N with 2-D VQ plus memory-6 and memory-8 shaping trellises. It is seen that the pdfs in both cases are quite far from being Gaussian. This indicates that the source code (or quantizer) in [10] is underperforming. As mentioned in Section II-C, Marcellin and Fischer [19] reported a granular gain of 1.33 dB with 256-state (or memory-8) TCQ, but 2-D VQ with memory-8 trellis shaping in [10] only realizes 1.28-dB shaping gain (even with specially designed generator polynomials for the shaping codes) at roughly the same complexity.

A closer examination of the source code in [10], which employs 2-D VQ followed by trellis shaping, reveals that it is in fact 2-D trellis-coded vector quantization (TCVQ) [12], [31]. This explains why the above 1.28-dB shaping gain is exactly the same achieved with 2-D TCVQ in [34] with a memory-8 trellis.

The basic idea of both TCQ and TCVQ is to allow an expanded signal set and use coded modulation [29] for set partitioning. They differ in the dimensionality of the codebook to start with (or before set partitioning). In TCQ [19], because each R-bit codeword consists of one trellis bit and $R-1$ codeword bits (see Section II-C), the rate of TCQ is at least 1 b/s. TCVQ [12], [31] was invented mainly to achieve fractional bit rates (per dimensional) in source coding.

The minimum Euclidean distances of the partitioned signal sets determine the distant spectrum of the resulting trellis modulation codes [29], and thus control the granular gain of the resulting TCQ or TCVQ to a great extent [34]. However, according to [31], "the gain due to partitioning in the sense of increasing the Euclidean distances between the codevectors is smaller in multidimensional spaces than that in the lower dimensional spaces." For example, as shown in Fig. 7, each time the TCQ codebook is partitioned into two subsets, the minimum Euclidean distance within each subset increases by a factor of two. But for 2-D TCVQ, the increasing factor of minimum distance at each partition is only $\sqrt{2}$. This is why the granular gain of 2-D TCVQ (e.g., 1.28 dB) is lower than that of TCQ (e.g., 1.33 dB) with the same memory (or number of states).

As will be explained shortly in Section IV-A, the role of quantization in DPC is to satisfy the power constraint and the aim is to shoot for high granular/shaping gain, the rate of the quantizer

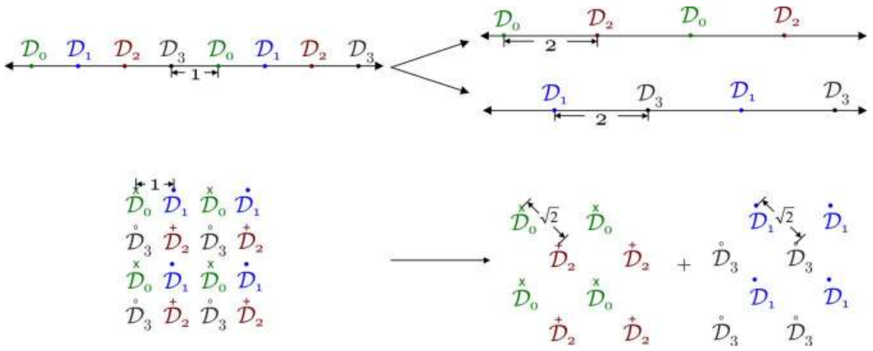
$\mathcal{D}_0$ $\mathcal{D}_1$ $\mathcal{D}_2$ $\mathcal{D}_3$ $\mathcal{D}_0$ $\mathcal{D}_1$ $\mathcal{D}_2$ $\mathcal{D}_3$
1
$\mathcal{D}_0$ $\mathcal{D}_2$ $\mathcal{D}_0$ $\mathcal{D}_2$
2
$\mathcal{D}_1$ $\mathcal{D}_3$ $\mathcal{D}_1$ $\mathcal{D}_3$
2
1
$\sqrt{2}$
+

Fig. 7. Set partition for TCQ (top) and TCVQ (bottom).

TABLE I
THE MODULO LOSS SNR_m FOR TCQ OF DIFFERENT NUMBER OF STATES AND THE PREDICTED TOTAL PERFORMANCE LOSS $\Delta\mathrm{SNR}$ IN OUR DIRTY-PAPER CODE DESIGNS, ASSUMING THE PACKING LOSS SNR_p FROM IRA CODES IS 0.20 dB AND A TARGET RATE OF $C^* = 0.25$ b/s. THE TCQ GENERATOR POLYNOMIALS (h_0, h_1) ARE TAKEN FROM [34]

# of states in TCQ	h_0 (in octal form)	h_1 (in octal form)	$g(\Lambda)$ (dB)	$\Delta\mathrm{SNR}_\mathrm{m}$ (dB)	$\Delta\mathrm{SNR}_\mathrm{p}$ (dB)	$\Delta\mathrm{SNR}$ (dB)
64	127	42	1.2698	0.8395	0.20	1.0395
128	233	156	1.3083	0.7234	0.20	0.9234
256	625	242	1.3428	0.6175	0.20	0.8175
512	1047	702	1.3662	0.5447	0.20	0.7447
1024	2463	1022	1.3863	0.4815	0.20	0.6815
2048	4261	3426	1.4032	0.4278	0.20	0.6278

is a nonfactor. We are thus motivated to seek for dirty-paper code designs using TCQ and IRA codes, since a design that employs TCQ instead of TCVQ will have a higher granular gain, hence, smaller modulo loss $\Delta\mathrm{SNR}_m$ in (10) and better performance. The use of TCQ is the main difference between our work and that of Erez and ten Brink [10], which in fact employs 2-D TCVQ via 2-D VQ plus trellis shaping for source coding. Another difference is that we combine TCQ and IRA codes without systematic doping. Details of our code designs are given next.

IV. DPC USING TCQ AND IRA CODES

In this section, we start with guidelines for near-capacity dirty-paper code designs, before detailing our proposed designs that combine TCQ with both nonsystematic and systematic IRA codes.

A. Design Guidelines

In [25], we offered an algebraic *message-based binning* interpretation of DPC in terms of source–channel coding. From an information-theoretical perspective, there are granular gain and boundary gain in source coding, and packing/coding gain[1] and shaping gain in channel coding. DPC is primarily a channel coding problem (for transmitting messages), one should consider the packing gain and the shaping gain. In addition, the side information necessitates source coding to satisfy the power constraint, i.e., the constellation needs to be infinitely replicated so that one can quantize the side information to satisfy the power constraint [38]. Thus, source coding in DPC is not conventional in the sense that there is only granular gain, but no boundary gain. One needs to establish the equivalence between the shaping gain in channel coding and the granular gain in source coding for DPC. Then one can shoot for the shaping gain via source coding and the packing gain via channel coding. In practice, the former should be done with quantizers (e.g., TCQ [19]) having almost spherical Voronoi cells in a high-dimensional Euclidean space, and the latter with near-capacity channel codes (e.g., IRA codes [18]). This justifies our combined source–channel coding approach to DPC based on TCQ and IRA codes.

Treating TCQ as an *equivalent* lattice VQ, we seamlessly combine TCQ with both nonsystematic and systematic IRA codes in a source–channel coding setup without sacrificing the performance of either component. That is, both the TCQ and IRA code components still work best when combined together in our elaborate dirty-paper code constructions. Because of this, the practical performance loss $\Delta\mathrm{SNR}$ (in decibels) in our designs is the sum of the packing loss $\Delta\mathrm{SNR}_\mathrm{p}$ due to IRA codes and the modulo loss $\Delta\mathrm{SNR}_\mathrm{m}$ in (10) due to TCQ, i.e.,

$$\Delta\mathrm{SNR} = \Delta\mathrm{SNR}_\mathrm{p} + \Delta\mathrm{SNR}_\mathrm{m}. \tag{13}$$

After using the generator polynomials in [34] to implement TCQ of different number of states and subsequently measuring the equivalent $g(\Lambda)$, we compute $\Delta\mathrm{SNR}_\mathrm{m}$ from (10) with $C^* = 0.25$ b/s. Assuming that $\Delta\mathrm{SNR}_\mathrm{p} = 0.20$ dB (confirmed in our simulations), Table I lists the predicted total performance loss $\Delta\mathrm{SNR}$ when the target rate is $C^* = 0.25$ b/s.

[1] We prefer to use packing gain under the context of source–channel coding.

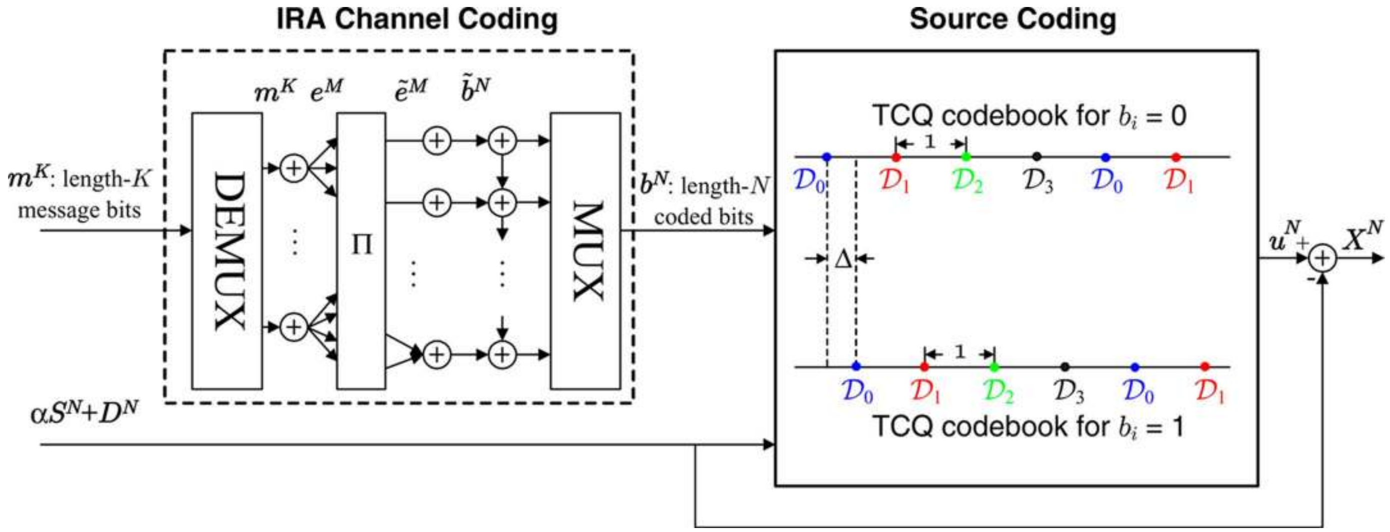
IRA Channel Coding
Source Coding
m^K: length-K message bits
DEMUX
m^K e^M
Π
$\tilde{e}^M$ $\tilde{b}^N$
MUX
b^N: length-N coded bits
$\alpha S^N + D^N$
TCQ codebook for $b_i = 0$
1
$\mathcal{D}_0$ $\mathcal{D}_1$ $\mathcal{D}_2$ $\mathcal{D}_3$ $\mathcal{D}_0$ $\mathcal{D}_1$
Δ
$\mathcal{D}_0$ $\mathcal{D}_1$ $\mathcal{D}_2$ $\mathcal{D}_3$ $\mathcal{D}_0$ $\mathcal{D}_1$
TCQ codebook for $b_i = 1$
u^N
+
-
X^N

Fig. 8. Block diagram of our proposed dirty-paper encoder based on TCQ and nonsystematic IRA code.

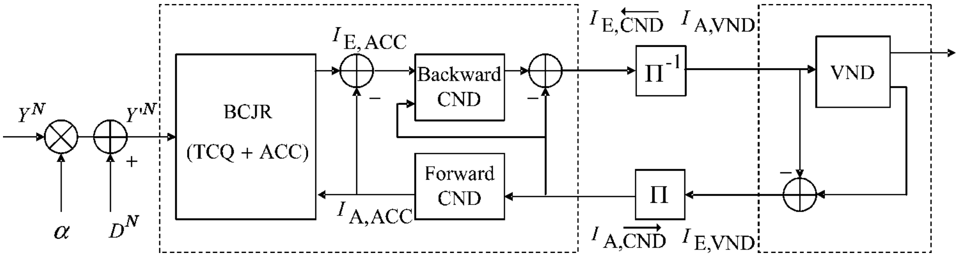

Fig. 9. Block diagram of the proposed decoder with TCQ and the nonsystematic IRA code.

Simulations show that the performance of our designs matches the predicted ones in Table I very well (Section V). Owing to the efficiency of TCQ and the simplicity of our combined source–channel coding approach, our code designs achieve better performance at lower complexity (see Section VII) than that in [10]. Our approach shares the same philosophy of separate source and channel coding in [2] via superposition coding. But our designs with 256-state TCQ perform better than the one in [2] with 512-state TCQ.

B. Proposed TCQ Plus Nonsystematic IRA Code Designs

Our proposed dirty-paper code design with TCQ and *nonsystematic* IRA code is shown in Fig. 8 (encoder) and Fig. 9 (decoder).

- **Encoding:** The input message is first converted to length-K message bits $m^K = (m_1, m_2, \ldots, m_K)^T \in \{0,1\}^K$, which is encoded by a rate-$\frac{K}{N}$ IRA code to length-N coded bits $b^N = (b_1, b_2, \ldots, b_N)^T \in \mathcal{B} \subset \{0,1\}^N$, where $\mathcal{B}$ is the set of all 2^K binary codewords of the IRA code. The IRA encoder consists of four parts.
  1) A *variable node encoder* that irregularly repeats each bit in m^K according to a variable node profile $\lambda(x) = \sum_{d=2}^{d_v} \lambda_d x^d$. The encoded M bits are denoted as $e^M = (e_1, e_2, \ldots, e_M)^T$, corresponding to the M edges entering the interleaver.
  2) An *interleaver* that randomly interleaves the input bits e^M to $\tilde{e}^M = (\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_M)^T = \Pi(e^M)$.
  3) A check node encoder that bi-regularly sums up the interleaved edge bits $\tilde{e}^M$ according to a check node profile $\rho(x) = \rho_1 x + \rho_{d_c} x^{d_c}$. $\rho(x)$ is chosen to be bi-regular with minimum check node degree of one so that decoding can start properly [10]. Denote the output length-N bits as $\tilde{b}^N$.
  4) An ACC that outputs the accumulated bits of $\tilde{b}^N$ as $b^N = (b_1, \ldots, b_N)^T$, i.e., $b_i = \mathrm{mod}(\sum_{j=1}^{i} \tilde{b}_j, 2)$.

Then each bit of b^N is used to select TCQ 1-D codebook from two candidates (corresponding to bit value "0" and "1," respectively), which are a relatively shifted version of each other with a tunable shift Δ (see Fig. 8). Equivalently, we write the resulting N-dimensional TCQ codebook as

$$\Lambda + \Delta \cdot b^N = \{4z^N + c^N + \Delta \cdot b^N : z^N \in \mathbb{Z}^N \text{ and } c^N \in \mathcal{C}\}. \tag{14}$$

Since $\Lambda + \Delta \cdot b^N$ is only a shifted version of the original TCQ codebook Λ, the normalized second moment of the Voronoi regions does not change. To encode, the side information $S^N \in \mathbb{R}^N$ is linearly scaled by α, dithered by a length-N sequence D^N that is uniformly distributed over the N-dimensional cube $[0,4)^N$, and then $\alpha S^N + D^N$ is quantized to u^N by the TCQ with codebook $\Lambda + \Delta \cdot b^N$ selected by codeword b^N so that the obtained quantization error $X^N = u^N - \alpha S^N - D^N$ satisfies the power constraint i$\frac{1}{N} E\|X^N\|^2 \leq P_X$. X^N is transmitted over the side-information channel.

- **Decoding:** The decoder receives $Y^N = X^N + S^N + Z^N$ and aims at finding the codeword
$$\hat{u}^N \in \Sigma \triangleq 4\mathbb{Z}^N \oplus \mathcal{C} \oplus \Delta\mathcal{B} = \bigcup_{b^N \in \mathcal{B}} (\Lambda + \Delta \cdot b^N) \quad (15)$$
that is closest to $Y'^N = \alpha Y^N + D^N = u^N + Z'^N$, where $Z'^N \triangleq (1-\alpha)(-X^N) + \alpha Z^N$ is the equivalent channel noise. Schematically, the decoder consists of an inner Bahl–Cocke–Jelinek–Raviv (BCJR) decoder based on TCQ and ACC, a forward CND, a backward CND, and a VND. Decoding starts at the BCJR decoder, which takes the received signal Y^N and *a priori* log-likelihood ratios (LLR) $L_{\text{ap,ACC}}(\tilde{b}_i)$ (initialized to zero before the first iteration) for each bit $\tilde{b}_i, i = 1, 2, \ldots, N$, and computes extrinsic LLRs $L_{\text{ext,ACC}}(\tilde{b}_i)$. We denote $\tilde{B}_i$ and $L_{\text{ap,ACC}}(\tilde{B}_i)$ as the corresponding random variables of $\tilde{b}_i$ and $L_{\text{ap,ACC}}(\tilde{b}_i)$, respectively, and use similar notations in the sequel. Define
$$I_{A,\text{ACC}} = \frac{1}{N}\sum_{i=1}^{N} I(L_{\text{ap,ACC}}(\tilde{B}_i), \tilde{B}_i) \quad (16)$$
$$I_{E,\text{ACC}} = \frac{1}{N}\sum_{i=1}^{N} I(L_{\text{ext,ACC}}(\tilde{B}_i), \tilde{B}_i) \quad (17)$$
as the *a priori* and extrinsic information at the input and output of the BCJR decoder, respectively. Then $L_{\text{ext,ACC}}(\tilde{b}_i)$ is fed to the backward CND as the *a priori* information. The backward CND generates extrinsic LLRs $L_{\text{ext},\overleftarrow{\text{CND}}}(\tilde{e}_i)$ for each interleaved edge bit $\tilde{e}_i, i = 1, 2, \ldots, M$, and $L_{\text{ext},\overleftarrow{\text{CND}}}(\tilde{e}_i)$ is forwarded through the edge deinterleaver Π^{-1} to the VND. The VND uses *a priori* LLRs
$$L_{\text{ap,VND}}(e_i) = \Pi^{-1}(L_{\text{ext},\overleftarrow{\text{CND}}}(\tilde{e}_i))$$
to generate the extrinsic LLRs $L_{\text{ext,VND}}(e_i)$ that is fed through the edge interleaver Π to the forward CND, i.e., $L_{\text{ap},\overrightarrow{\text{CND}}}(\tilde{e}_i) = \Pi(L_{\text{ext,VND}}(e_i))$. One decoding iteration is completed by feeding back the extrinsic LLRs $L_{\text{ext},\overrightarrow{\text{CND}}}(\tilde{b}_i)$ generated by the forward CND to the BCJR decoder such that $L_{\text{ap,ACC}}(\tilde{b}_i) = L_{\text{ext},\overrightarrow{\text{CND}}}(\tilde{b}_i)$. Decoding then proceeds to the next iteration until the hard decisions of the information bits m^K at the VND remain unchanged after a certain number of iterations (e.g., 10). To design the variable and check node profiles of the IRA code, the following quantities are crucial:
$$I_{A,\text{VND}} = \frac{1}{M}\sum_{i=1}^{M} I(L_{\text{ap,VND}}(E_i), E_i)$$
$$I_{E,\text{VND}} = \frac{1}{M}\sum_{i=1}^{M} I(L_{\text{ext,VND}}(E_i), E_i) \quad (18)$$
$$I_{A,\overrightarrow{\text{CND}}} = \frac{1}{M}\sum_{i=1}^{M} I(L_{\text{ap},\overrightarrow{\text{CND}}}(\tilde{E}_i), \tilde{E}_i)$$
$$I_{E,\overleftarrow{\text{CND}}} = \frac{1}{M}\sum_{i=1}^{M} I(L_{\text{ext},\overleftarrow{\text{CND}}}(\tilde{E}_i), \tilde{E}_i) \quad (19)$$
i.e., the *a priori* and extrinsic information at the input and output of the VND and the forward and backward CNDs, respectively.

C. Proposed TCQ Plus Systematic IRA Code Designs

Our proposed dirty-paper code design with TCQ and *systematic* IRA code is shown in Fig. 10 (encoder) and Fig. 11 (decoder).

- **Encoding:** The input message m^K is encoded by a rate-$\frac{K}{N}$ systematic IRA code to length-N coded bits
$$((m^K)^T, (b^{N-K})^T)^T = (m_1, m_2, \ldots, m_K, b_1, b_2, \ldots, b_{N-K})^T$$
where m^K is the systematic part, and b^{N-K} is the parity part. Then m^K and b^{N-K} are used to select a K-dimensional and an $(N-K)$-dimensional TCQ codebook from TCQ$_2$ and TCQ$_1$, respectively (see Fig. 10). The resulting two TCQ codebooks, denoted as $\Lambda_2 + \Delta_2 \cdot m^K$ and $\Lambda_1 + \Delta_1 \cdot b^{N-K}$, are used to quantize the first K samples and the remaining $N-K$ samples of $\alpha S^N + D^N$, respectively, so that the obtained quantization error X^N satisfies $\frac{1}{N}E\|X^N\|^2 \le P_X$. Then X^N is transmitted over the side-information channel.
- **Decoding:** As shown in Fig. 11, two BCJR decoders are used. The first one is for the parity part and computes $L_{\text{ext,ACC}}(\tilde{b}_i), i = 1, 2, \ldots, N-K$, while the second is for the systematic part and computes $L_{\text{ext,TCQ}_2}(m_j)$, $j = 1, 2, \ldots, K$. The backward CND, deinterleaver, forward CND, and interleaver implement the same functions as those in our TCQ and nonsystematic IRA code design. The main difference is in the VND, where the extrinsic LLRs are computed as
$$L_{\text{ext,VND}}(e_i) = L_{\text{ext,TCQ}_2}(m_j) + \sum_{e \in \mathcal{E}_j \setminus i} L_{\text{ap,VND}}(e), \quad i = 1, 2, \ldots, M \quad (20)$$
where j is the index of the variable node that connects to edge e_i, and $\mathcal{E}_j \setminus i$ is the indices of the edges other than i itself that connects to the same variable node j.

In general, TCQ$_2$ in the systematic part can be different from TCQ$_1$ in the parity part as long as they are designed such that both satisfy the same power constraint P_X. This way, they can be used interchangeably from a power consumption point of view. However, the difference between them lies in their granular gain. Since the decoding complexity for TCQ$_2$ is lower than that for TCQ$_1$, within the same complexity limit, the flexibility offered by two TCQs and the potential for achieving higher granular gains than a single TCQ can (with nonsystematic IRA codes) are our main impetuses for studying code designs based on systematic IRA codes.

D. Tuning Δ for the Best Performance–Complexity Tradeoff

One of the key advantage of our dirty-paper code designs with TCQ and IRA codes compared to the design in [10] is that the

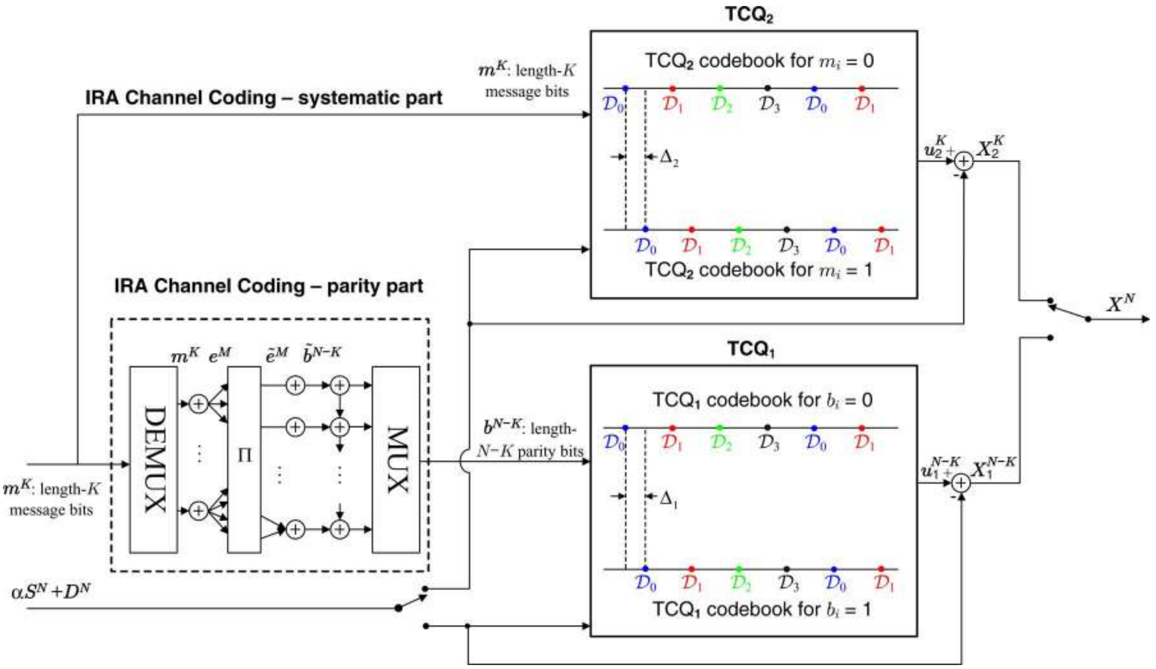
TCQ₂
TCQ₂ codebook for $m_i = 0$
$\mathcal{D}_0$ $\mathcal{D}_1$ $\mathcal{D}_2$ $\mathcal{D}_3$ $\mathcal{D}_0$ $\mathcal{D}_1$
Δ_2
u_2^K
X_2^K
TCQ₂ codebook for $m_i = 1$
m^K: length-K message bits
IRA Channel Coding – systematic part
IRA Channel Coding – parity part
m^K e^M $\tilde{e}^M$ $\tilde{b}^{N-K}$
DEMUX
Π
MUX
b^{N-K}: length-$N-K$ parity bits
m^K: length-K message bits
$\alpha S^N + D^N$
X^N
TCQ₁
TCQ₁ codebook for $b_i = 0$
Δ_1
u_1^{N-K}
X_1^{N-K}
TCQ₁ codebook for $b_i = 1$

Fig. 10. Block diagram of the proposed dirty-paper encoder based on TCQ and the systematic IRA code.

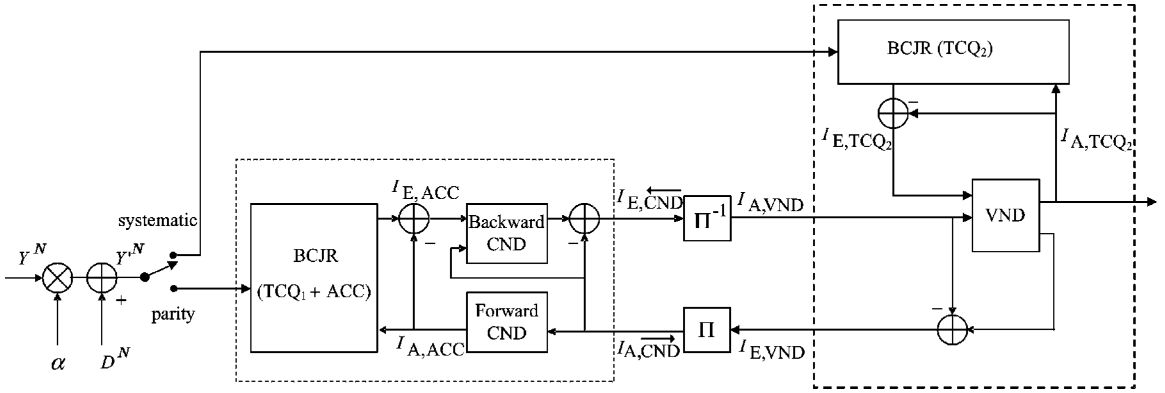

Fig. 11. Block diagram of the decoder with TCQ and the systematic IRA code.

shift Δ in our designs can be continuously tuned for the best performance–complexity tradeoff, where the performance is measured by the capacity of the TCQ-induced channel, and the complexity is interpreted as the y-intercept of the ACC EXIT chart (since the higher the y-intercept, the easier the code design).

To highlight this advantage, we first give a simple example with a 4-state length-2 TCQ and a rate-$\frac{1}{2}$ channel code, i.e., $N = 2$, $K = 1$, the TCQ codebook is $\Lambda = 4\mathbb{Z}^2 \oplus \mathcal{C}$ with

$$\mathcal{C} = \{(0,0)^T, (0,2)^T, (2,1)^T, (2,3)^T\}$$

as in Fig. 5, and the channel codebook $\mathcal{B}$ contains two binary vectors from the set $\{(0,0)^T, (0,1)^T, (1,0)^T, (1,1)^T\}$. Then the set of all possible TCQ codewords can be written as $\Sigma = 4\mathbb{Z}^2 \oplus \mathcal{C} \oplus \Delta\mathcal{B}$. Since the block length is fixed at two, unbounded channel noise will result in a zero channel capacity. Hence, we assume that the equivalent channel noise is uniformly distributed over a disk $\mathcal{O}(r) = \{z'^2 \in \mathbb{R}^2 : z_1'^2 + z_2'^2 \le r^2\}$ and use the minimum distance $d_{\min}$ of the channel codebook Σ as an alternative measure for the capacity of the TCQ-induced channel. Then it is easy to verify that the largest minimum distance $d_{\min} = \frac{5\sqrt{2}}{6}$ is achieved when $\Delta = \frac{5}{6}$ and $\mathcal{B} = \{(0,0)^T, (1,1)^T\}$, i.e., the optimal channel code is a length-2 repetition code in this example.

Fig. 12 compares the Voronoi regions of Σ for $\Delta = 1$ and $\Delta = \frac{5}{6}$ (with the same length-2 repetition code), where the crosses are TCQ codewords from Λ, the dots are those from $\Lambda + (\Delta, \Delta)^T$, the solid lines are the boundary of the Voronoi regions of Σ, and the dashed lines are the boundary of the Voronoi regions of Λ. We can see that $\Delta = \frac{5}{6}$ achieves a larger minimum distance than $\Delta = 1$. In summary, although this example assumes a slightly different channel model, it illustrates the main idea of tuning Δ for the best channel code performance.

Now consider our TCQ plus nonsystematic IRA code designs. In a similar manner as the above example, we fix the TCQ codebook Λ, the target rate $\frac{K}{N}$, and the channel SNR, and search for the optimal Δ, where the optimization criterion is to maximize the area under the $(I_{A,\mathrm{ACC}}, I_{E,\mathrm{ACC}})$ EXIT curve, which is shown to be equal to the capacity of the corresponding channel [1] even if the channel has memory (such as the TCQ-induced channel we are designing IRA codes for).

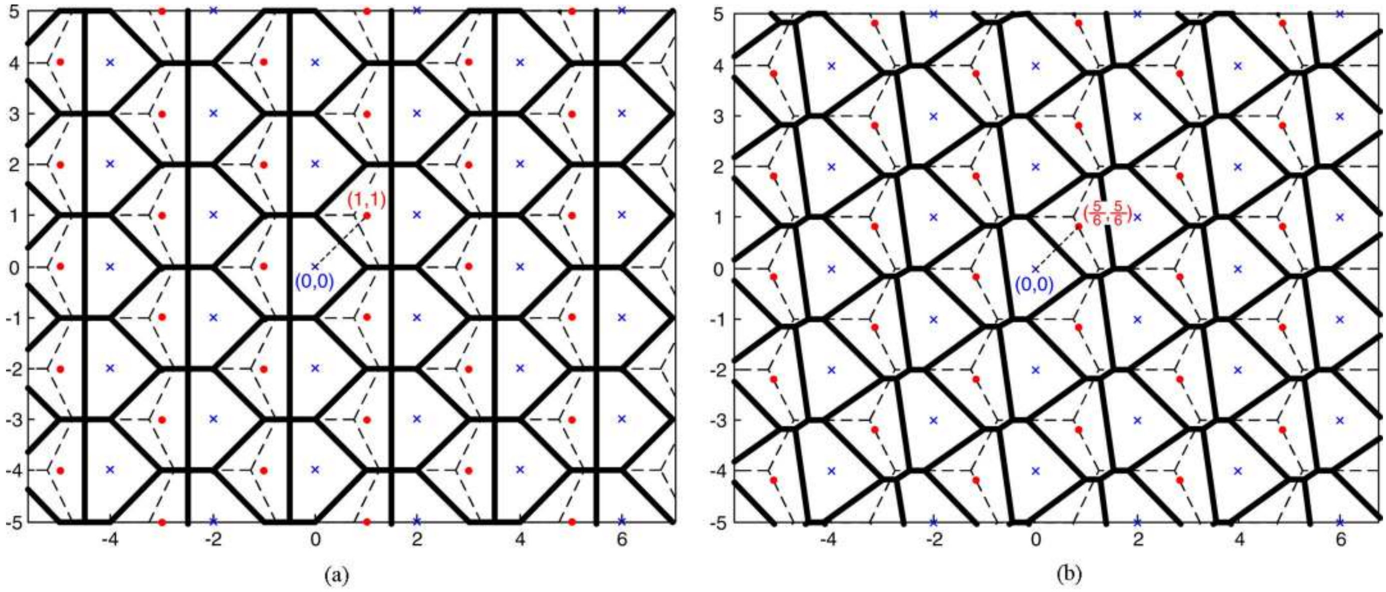

Fig. 12. Voronoi regions of the channel codebook Σ with (a) $\Delta = 1.0, d_{\min} = 1$ and (b) $\Delta = \frac{5}{6}, d_{\min} = \frac{5\sqrt{2}}{6} \approx 1.18$ based on the same length-2 repetition code and 4-state length-2 TCQ. Crosses are TCQ codewords from Λ and dots represent $\Lambda + (\Delta, \Delta)^T$. Dashed lines are the boundary of the Voronoi regions of Λ.

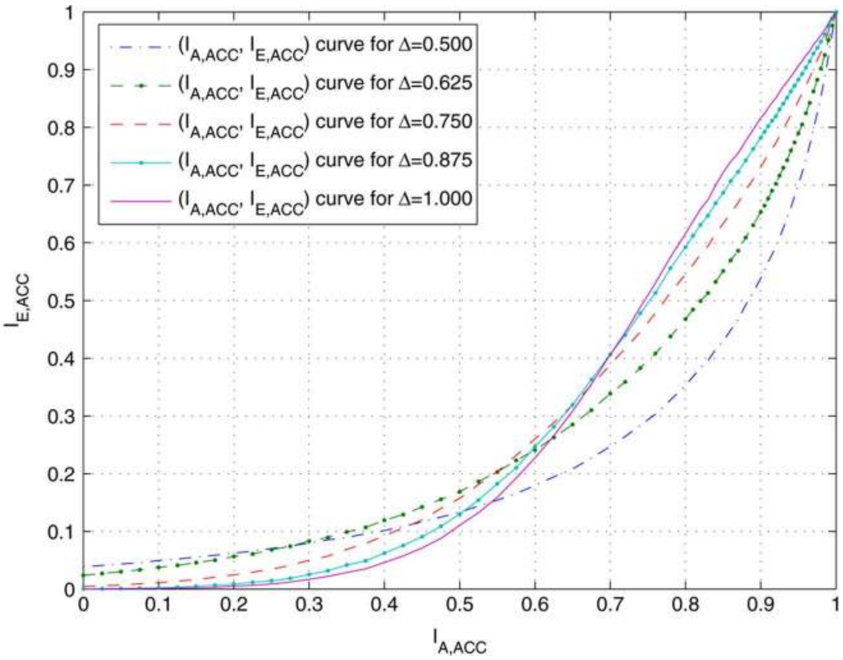

Fig. 13. $(I_{A,\mathrm{ACC}}, I_{E,\mathrm{ACC}})$ EXIT curves for 64-state TCQ with various Δ values in $[0.5, 1]$. $\frac{K}{N} = 0.25$.

Fig. 13 plots the $(I_{A,\mathrm{ACC}}, I_{E,\mathrm{ACC}})$ EXIT curves for 64-state TCQ with $\frac{K}{N}$ fixed at 0.25 and various Δ values in $[\frac{q}{2}, q] = [0.5, 1]$. We observe that as Δ approaches one from below, the beginning part of the EXIT curve becomes lower, while the ending part gets higher. Interestingly, as Δ increases in the range $[0.5, 1]$, the area under the $(I_{A,\mathrm{ACC}}, I_{E,\mathrm{ACC}})$ EXIT curve reaches the highest point at $\Delta \approx 0.8$, where the y-intercept is 0.001783. More numbers are given in Table II for 4-, 16-, 64-, 256-, 512-, 1024-, and 2048-state TCQs.

Fig. 14 replots the data in Fig. 13 to show the tradeoff, as Δ varies, between the area under (or approximate capacity of the TCQ-induce channel) and the y-intercept of the $(I_{A,\mathrm{ACC}}, I_{E,\mathrm{ACC}})$ EXIT curves. Clearly, if one targets at maximizing the capacity, the optimal Δ should be chosen to maximize the area under the $(I_{A,\mathrm{ACC}}, I_{E,\mathrm{ACC}})$ EXIT curve. However, such an optimal Δ might lead to an extremely small y-intercept, which makes IRA code design too difficult.

In our code designs, we choose Δ such that the approximated capacity is maximized under the constraint that the y-intercept is lower-bounded by a fixed number $\eta = 0.001$, which is carefully chosen such that IRA code design has manageable complexity while the resulting optimal Δ corresponds to an approximated capacity that is very close to its maximum. Based on the data in Table II, this criterion gives us optimal $\Delta = 0.95, 0.85, 0.80, 0.75, 0.70, 0.70, 0.70$ for 4-, 16-, 64-, 256-, 512-, 1024-, and 2048-state TCQ, respectively.

TABLE II
THE TRADEOFF BETWEEN THE AREA UNDER AND THE y-INTERCEPT OF THE $(I_{A,\text{ACC}}, I_{E,\text{ACC}})$ EXIT CURVES

Δ / # of states		0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
4	area	0.267217	0.277934	0.284838	0.288568	0.290406	0.291154	**0.291438**	0.291545
	y-int.	0.050004	0.044182	0.034985	0.023686	0.013814	0.005963	**0.001387**	0.000000
16	area	0.250006	0.255998	0.258921	0.260290	**0.260764**	0.260751	0.260618	0.260553
	y-int.	0.025260	0.017866	0.011243	0.006089	**0.002695**	0.000965	0.000192	0.000000
64	area	0.267168	0.271501	0.273149	**0.273458**	0.273364	0.273118	0.272645	0.272489
	y-int.	0.019000	0.010249	0.004544	**0.001783**	0.000666	0.000194	0.000021	0.000000
256	area	0.269911	0.272458	**0.272991**	0.272787	0.272463	0.272213	0.271666	0.271529
	y-int.	0.012512	0.004627	**0.001342**	0.000342	0.000088	0.000021	0.000000	0.000000
512	area	0.268387	**0.270324**	0.270333	0.270042	0.269688	0.269347	0.268736	0.268467
	y-int.	0.009401	**0.002981**	0.000766	0.000176	0.000020	0.000002	0.000001	0.000000
1024	area	0.268879	**0.270315**	0.270696	0.270365	0.269760	0.269236	0.268485	0.268083
	y-int.	0.007329	**0.001913**	0.000425	0.000070	0.000016	0.000003	0.000000	0.000000
2048	area	0.268859	**0.271061**	0.270972	0.270360	0.269501	0.268857	0.268450	0.268268
	y-int.	0.005537	**0.001208**	0.000185	0.000031	0.000006	0.000000	0.000000	0.000000

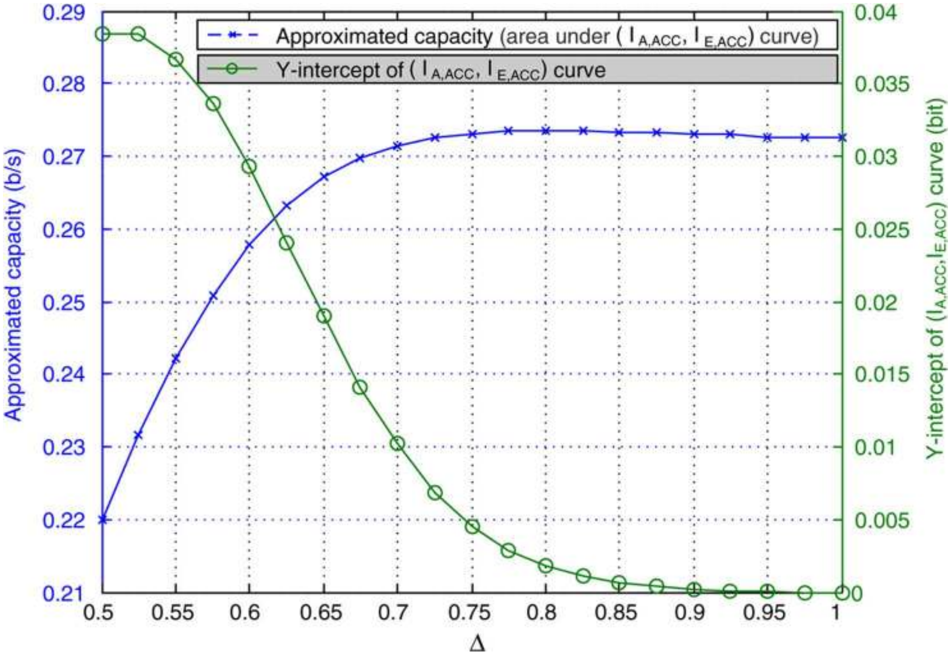

Fig. 14. Δ controls the tradeoff between the approximated capacity and the y-intercept of the $(I_{A,\text{ACC}}, I_{E,\text{ACC}})$ EXIT curves.

V. SIMULATION RESULTS WITH TCQ AND NONSYSTEMATIC IRA CODES

In our implementations, to achieve the target rate of 0.25 b/s, we set $\frac{K}{N} = \frac{1}{4}$ with $K = 120\,000$ and $N = 480\,000$ bits, and use the optimal Δ's shown in Table II for 256-, 1024-, and 2048-state TCQs.

For various values of ρ_1 (i.e., the fraction of degree-1 check node) and d_c (i.e., the maximum check node degree), we examine the $(I_{A,\overrightarrow{\text{CND}}}, I_{E,\overleftarrow{\text{CND}}})$ EXIT charts and select the best CND profiles as shown in Table III. The VND EXIT chart starts from the origin, so there is a small *vertical* opening between the starting points of the CND and VND EXIT charts for practical IRA code design.

Based on the VND EXIT charts with different variable node degrees (e.g., dashed curves in Fig. 15 with 256-state TCQ), we design the VND profile by matching the EXIT chart of the $(I_{A,\overrightarrow{\text{CND}}}, I_{E,\overleftarrow{\text{CND}}})$ part. Figs. 16, 17, and 18 show the EXIT charts of the nonsystematic IRA codes with 256-, 1024-, and 2048-state TCQ, respectively. It is seen that in each figure there is a tunnel between the $(I_{A,\overrightarrow{\text{CND}}}, I_{E,\overleftarrow{\text{CND}}})$ and $(I_{A,\text{VND}}, I_{E,\text{VND}})$ EXIT charts, which enables convergence.

We point out that the y-intercept in Table II is for the ACC EXIT curve, on which we make sure that the y-intercept is lower-bounded by $\eta = 0.001$, while all the EXIT curves presented in Figs. 16–18 are for CND. However, with the same CND profile, the y-intercept of the CND EXIT curve is a

TABLE III
CODE PROFILES AND SIMULATION RESULTS OF DIRTY-PAPER CODE DESIGNS BASED ON TCQ AND NONSYSTEMATIC IRA CODES

# of states in TCQ	256	1024	2048
Optimal Δ	0.75	0.70	0.70
CND profile $\rho(x)$	$0.60x + 0.40x^2$	$0.60x + 0.40x^2$	$0.60x + 0.40x^2$
VND profile $\lambda(x)$	$0.3889x^2 + 0.4858x^3$ $+0.1005x^{10} + 0.0110x^{30}$ $+0.0126x^{75} + 0.0012x^{891}$	$0.2988x^2 + 0.5754x^3$ $+0.1065x^{11} + 0.0098x^{35}$ $+0.0070x^{75} + 0.0026x^{470}$	$0.2717x^2 + 0.6290x^3$ $+0.0687x^{10} + 0.0193x^{22}$ $+0.0094x^{100} + 0.0019x^{600}$
Predicted gap	0.8175 dB	0.6815 dB	0.6278 dB
Simulated SNR	-3.008 dB	-3.138 dB	-3.198 dB
# of iterations	90 - 190	80 - 190	120 - 190
# of simulations	20	10	10
BER	7.08×10^{-6}	8.33×10^{-7}	5.83×10^{-6}
Gap to SNR*	0.820 dB	0.690 dB	0.630 dB

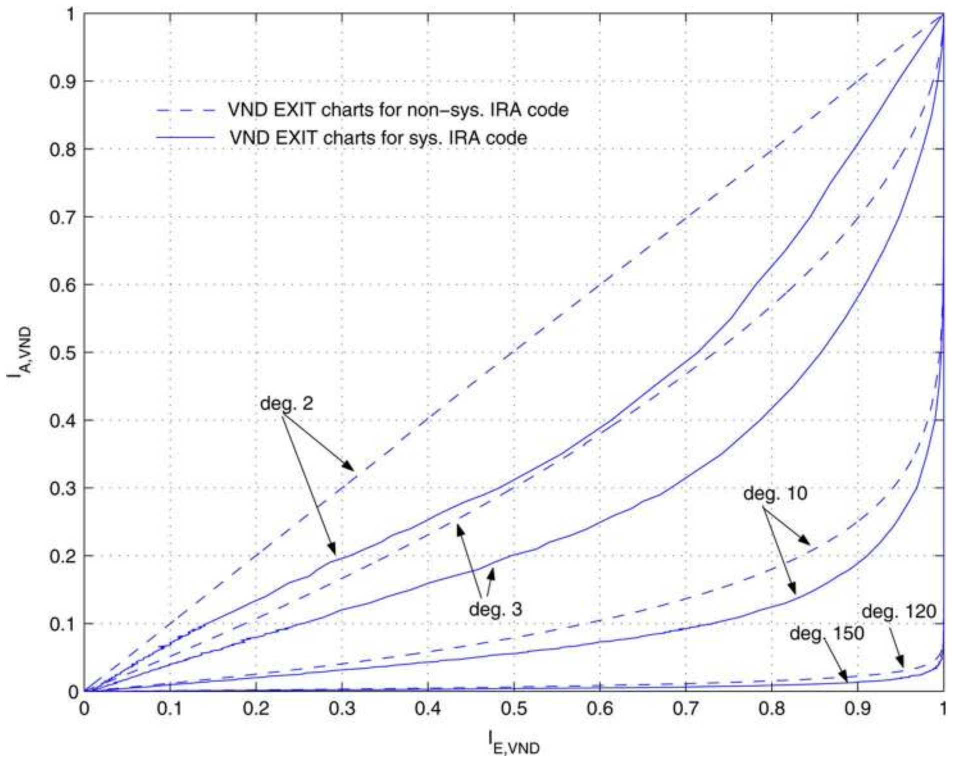

Fig. 15. VND EXIT charts with different variable node degrees. TCQ has 256 states.

monotonic function of the y-intercept of the ACC EXIT curves. For example, according to Table II, with 256-, 1024-, and 2048-state TCQ, the y-intercept of the ACC EXIT curve is $0.001342, 0.001913$, and 0.001208, respectively; while the y-intercept of the CND EXIT curve in Figs. 16–18 is approximately $0.0006, 0.0009$, and 0.0006, respectively.

With 256-state TCQ and the corresponding nonsystematic IRA code, after simulating 20 blocks of transmission, we obtain a bit-error rate (BER) of 7.08×10^{-6} when $\mathrm{SNR} = -3.008$ dB, which is 0.820 dB away from SNR^* for $C^* = 0.25$ b/s. This performance is almost the same as the predicted 0.8175-dB gap shown in Table I when the granular gain of TCQ is 1.3428 dB (again the 0.20-dB packing loss is due to practical IRA coding). The maximal number of decoding iterations is set to be 200 and the actual number to achieve convergence is between 90 and 190.

With 1024-state TCQ and the corresponding IRA code, after simulating 10 blocks of transmission, we obtain a BER of 8.33×10^{-7} when $\mathrm{SNR} = -3.138$ dB, which is 0.690 dB away from SNR^* at 0.25 b/s, matching the predicted performance in Table I. The maximal number of decoding iterations is set to be 200 and the actual number to achieve convergence is between 80 and 190.

With 2048-state TCQ and the corresponding IRA code, after simulating 10 blocks of transmission with our design based on the 2048-state TCQ, we obtain a BER of 5.83×10^{-6} when

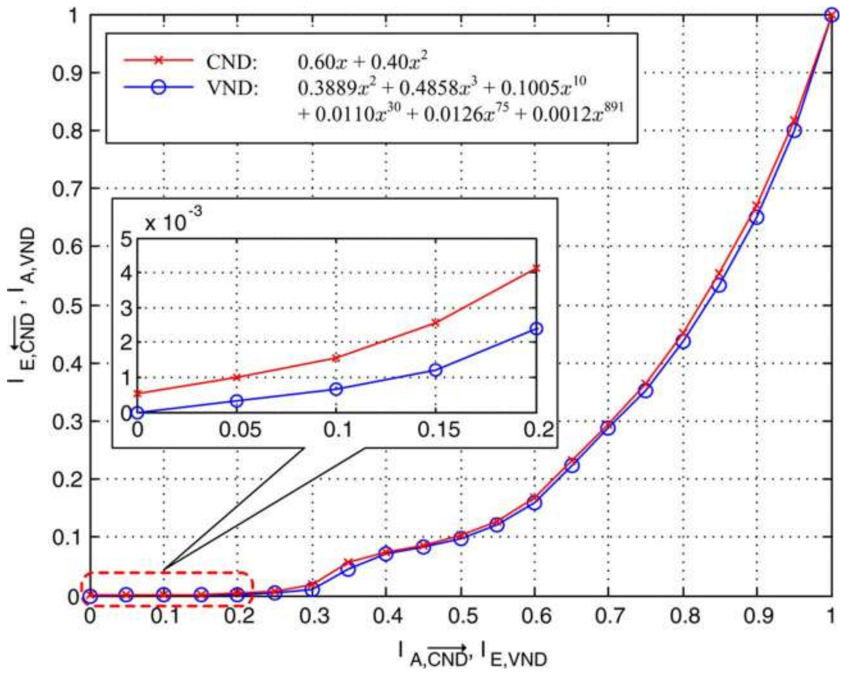

Fig. 16. EXIT charts of the nonsystematic IRA code at $\mathrm{SNR} = -3.008$ dB. TCQ has 256 states.

TABLE IV
CODE PROFILES AND SIMULATION RESULTS OF DIRTY-PAPER CODE DESIGNS BASED ON TCQ AND SYSTEMATIC IRA CODES

# of states in TCQ_1 & TCQ_2	256 & 256	1024 & 2048
Optimal Δ_1 & Δ_2	0.75 & 0.75	0.70 & 0.70
CND profile $\rho(x)$	$0.7333x + 0.2667x^4$	$0.7333x + 0.2667x^4$
VND profile $\lambda(x)$	$0.2104x^2 + 0.6857x^3$ $+0.0854x^8 + 0.0062x^{70}$ $+0.0114x^{100} + 0.0009x^{748}$	$0.2826x^2 + 0.4435x^3$ $+0.2593x^6 + 0.0144x^{120}$ $+0.0003x^{800}$
Predicted gap	0.8175 dB	0.6682 dB
Simulated SNR	-2.998 dB	-3.138 dB
# of iterations	100 - 140	140 - 190
# of simulations	20	10
BER	8.33×10^{-6}	9.17×10^{-6}
Gap to SNR*	0.830 dB	0.690 dB

$\mathrm{SNR} = -3.198$ dB, which is 0.630 dB away from the capacity in (3) at 0.25 b/s. This performance also matches the predicted 0.6278-dB gap in Table I when the granular gain of TCQ is 1.3863 dB. The maximal number of decoding iterations is set to be 200 and the actual number to achieve convergence is between 120 and 190.

The above results are better than our preliminary ones published in [24] at the same complexity. For example, with 1024-state TCQ, the proposed design has a $\Delta\mathrm{SNR}$ of 0.690 dB, which is smaller than that of 0.834 dB reported in [24]. This improvement is due to the increase in block length (from $N = 90\,000$ to $N = 480\,000$) and the optimized Δ (as opposed to using two TCQ codebooks that are shifted by a fixed amount of $\Delta = \frac{3q}{4} = 0.75$ in [24]).

VI. SIMULATION RESULTS WITH TCQ PLUS SYSTEMATIC IRA CODES

In our implementations, we set $\frac{K}{N} = \frac{1}{4}$ with $K = 120\,000$ and $N = 480\,000$ bits so that the rate $\frac{K}{N} = 0.25$ b/s. We consider two setups: one employs 256-state TCQ_1 and TCQ_2 (with $\Delta_1 = \Delta_2 = 0.75$ and an individual/overall granular gain of 1.3428 dB); another uses a 1024-state TCQ_1 (with $\Delta_1 = 0.70$ and 1.3863-dB granular gain) and a 2048-state TCQ_2 (with $\Delta_2 = 0.70$ and 1.4032-dB granular gain) for an overall granular gain of 1.3905 dB in source coding.[2] Figs. 19 and 20 show

[2] We directly use the optimal Δ's obtained in the design with nonsystematic IRA codes. Optimizing Δ_1 and Δ_2 might slightly improve the coding performance of our DPC code design with TCQ and systematic IRA codes. We leave this as a future research topic.

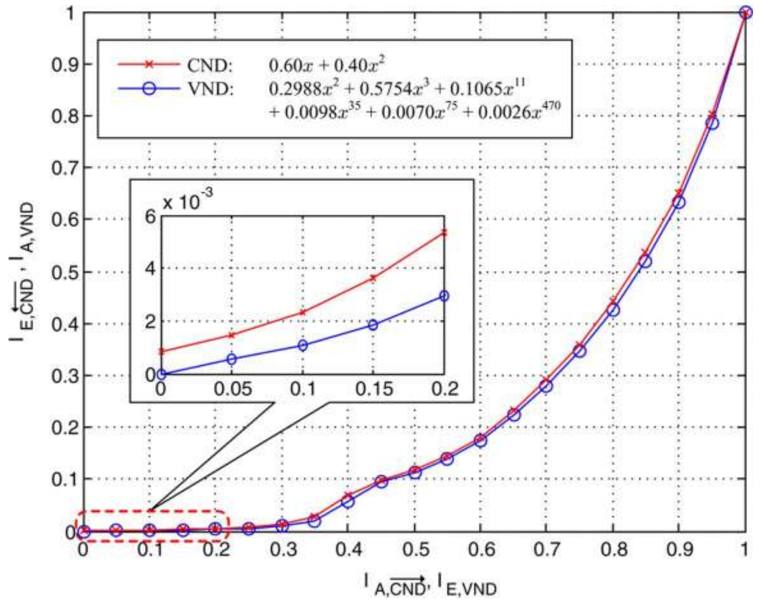

Fig. 17. EXIT charts of the nonsystematic IRA code at $\mathrm{SNR} = -3.138$dB. TCQ has 1024 states.

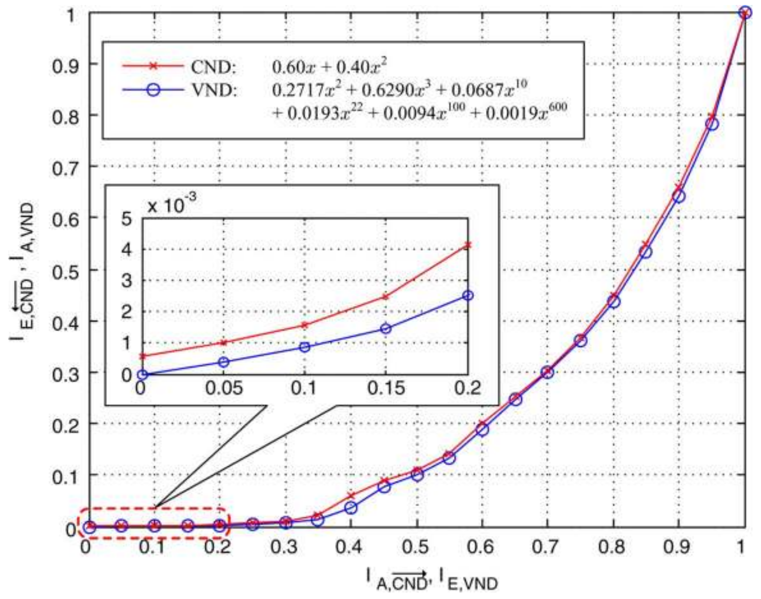

Fig. 18. EXIT charts of the nonsystematic IRA code at $\mathrm{SNR} = -3.198$ dB. TCQ has 2048 states.

the corresponding EXIT charts. In each setup, we compare the $(I_{A,\overrightarrow{\mathrm{CND}}}, I_{E,\overleftarrow{\mathrm{CND}}})$ EXIT charts and select the CND profiles as shown in Table IV. We design the VND profile by matching the EXIT chart of the $(I_{A,\overrightarrow{\mathrm{CND}}}, I_{E,\overleftarrow{\mathrm{CND}}})$ part. The resulting VND profiles are also given in the table.

In the setup with 256-state TCQ_1 and TCQ_2, we simulate 20 blocks of transmission at 0.25 b/s (as in the case with nonsystematic IRA codes and 256-state TCQ), and obtain a BER of 8.33×10^{-6} at the 0.830-dB gap from SNR^*. The maximal number of decoding simulations is again 200 and the actual number to achieve convergence is between 100 and 140.

In the setup with 1024-state TCQ_1 and a 2048-state TCQ_2, 10 blocks of transmission are simulated at 0.25 b/s. We obtain a BER of 9.17×10^{-6} when $\mathrm{SNR} = -3.138$ dB, which is 0.690 dB away from SNR^* at 0.25 b/s. The maximal number of decoding iterations is set to be 200 and the actual number to achieve convergence is between 140 and 190.

VII. Computational Complexity

Because the main computational complexity of our dirty-paper codes lies in BCJR decoding, we provide a quantitative complexity comparison between our designs and the one in [10] in terms of the number of *loop* operations needed for each BCJR decoding iteration. We assume a memory-8 VQ in [10], which performs 1.15 dB away from the capacity in (3) at 0.25 b/s (note again that the reported performance is 1.32 dB away from the capacity with a memory-6 VQ).

For the design in [10], $K = 60\,000$ and $N = 360\,000$ bits. The ACC takes 3-bit inputs with systematic doping applied to

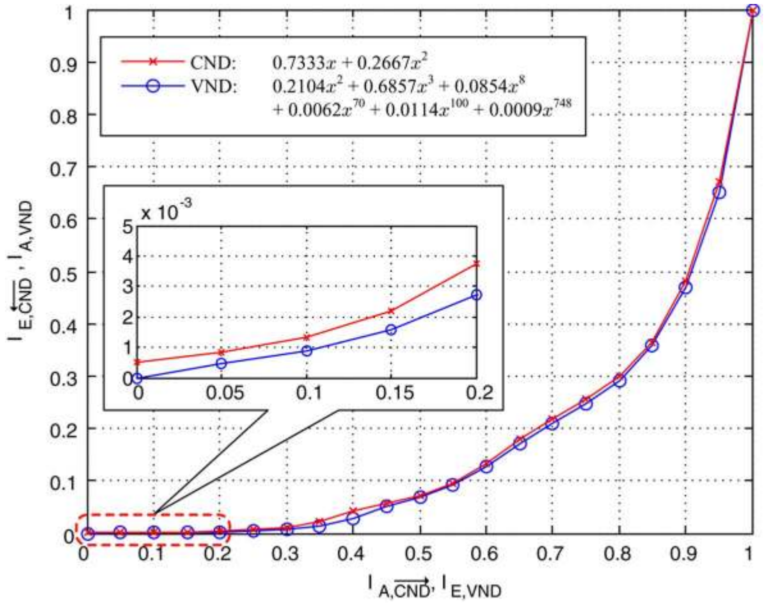

Fig. 19. EXIT charts of the systematic IRA code at $\mathrm{SNR} = -2.998$ dB. Both TCQ_1 and TCQ_2 have 256 states.

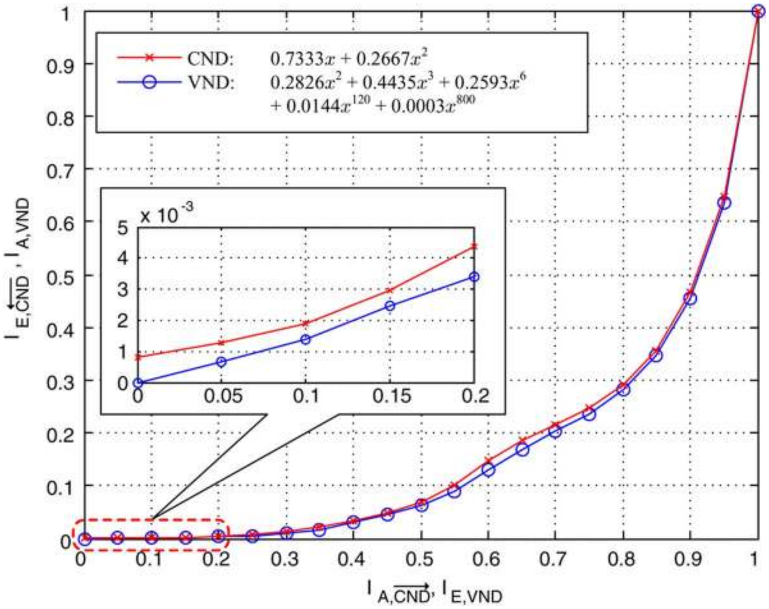

Fig. 20. EXIT charts of the systematic IRA code at $\mathrm{SNR} = -3.138$ dB. TCQ_1 has 1024 states and TCQ_2 has 2048 states.

the second and third bits, so the *effective* block length of TCQ is $N/3 = 120\ 000$. The number of input bits in the trellis is $1(\mathrm{VQ}) + 3(\mathrm{ACC}) = 4$. Thus, the number of loop operations is $2^{(8+1)}(\text{number of states}) \times 2^4(\text{input bits}) \times 120\ 000(\text{length}) = 512 \times 4 \times 480\ 000$ per BCJR decoding iteration.

In our design based on the nonsystematic IRA code and 256-state TCQ, which performs 0.82 dB away from the capacity in (3) at 0.25 b/s, $K = 120\ 000$, $N = 480\ 000$ bits, the number of input bits is $1(\mathrm{TCQ}) + 1(\mathrm{ACC}) = 2$. The BCJR decoder thus needs to run $2^{(8+1)} \times 2^2 \times 480\ 000 = 512 \times 4 \times 480\ 000$ loop operations per iteration. This design thus has the same complexity as that in [10]. We see that, although systematic (or unsigned bit) doping in [10], makes IRA code design an easier task, it exponentially increases the complexity of the BCJR decoder due to an increased number of input bits to the ACC. More importantly, when it is coupled with VQ, as manifested by the non-Gaussian pdf of the quantization error in Fig. 6, the source coding performance is degraded. In contrast, the pdf in Fig. 21 of the quantization error with the 256-state TCQ, in our code design without unsigned bit doping, is very close to be Gaussian. Thus, our design philosophy is much simpler with standard TCQ and no unsigned bit doping or tailor-designed generator polynomials for shaping codes, leading to better performance at the same complexity (or lower complexity if 128- or 64-state TCQ is used in our design, as seen from Table I).

In our design based on the systematic IRA code and 256-state TCQs, $K = 120\ 000$ and $N - K = 360\ 000$, the number of input bits is $1(\mathrm{TCQ}_2) + 1(\text{systematic bit}) = 2$ for the sys-

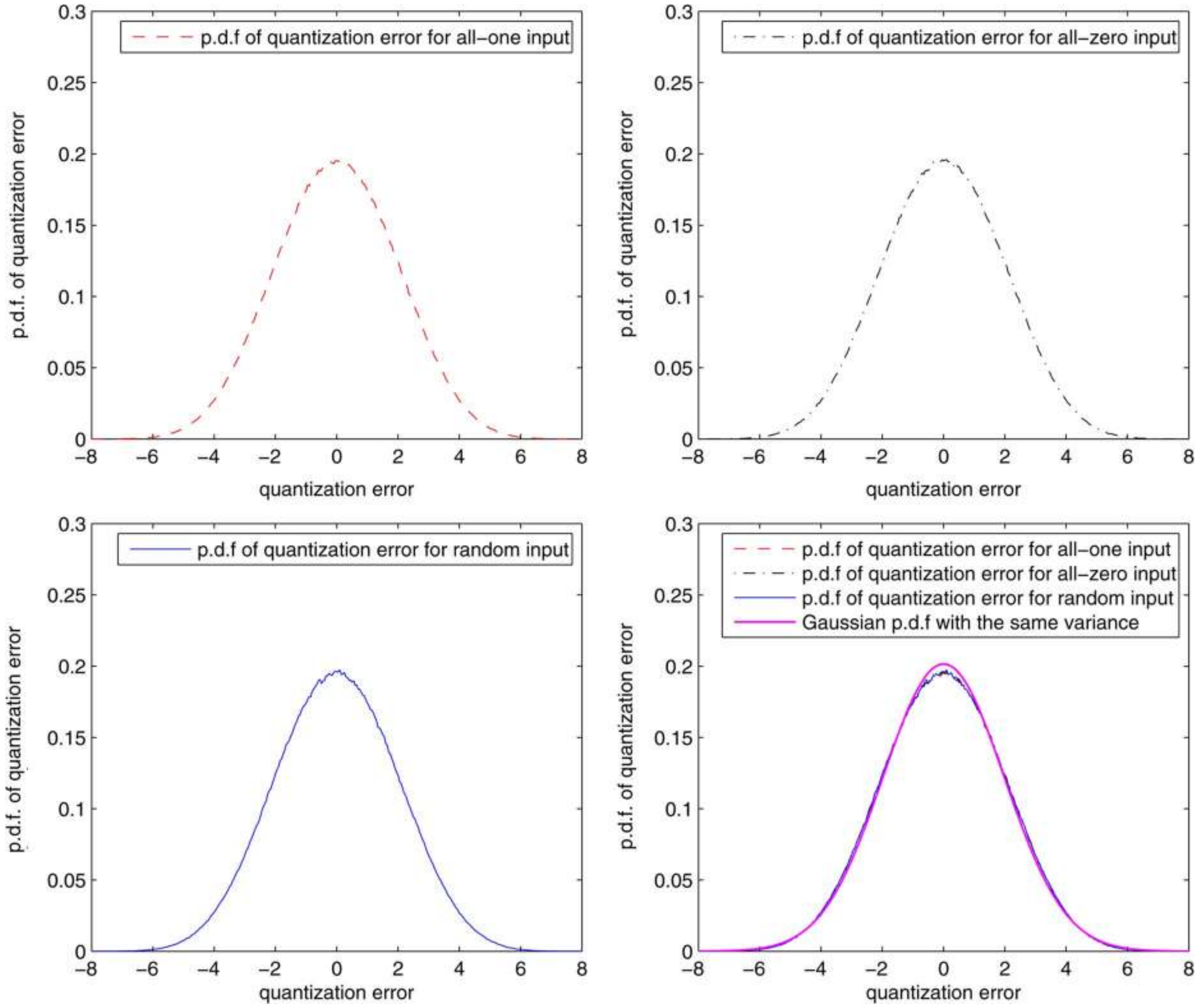

Fig. 21. The pdf of quantization error X for 256-state TCQ (with information bits m being all-one, all-zero, and binary random) in our dirty-paper code designs, together with a Gaussian pdf.

tematic part and $1(\mathrm{TCQ}_1) + 1(\mathrm{ACC}) = 2$ for the parity part. In the systematic part, the total number of loop operations is $2^8 \times 2^2 \times 120\,000 = 256 \times 4 \times 120\,000$. In the parity part, the number is $2^{(8+1)} \times 2^2 \times 360\,000 = 512 \times 4 \times 360\,000$. The combined number of loop operations is thus $512 \times 4 \times 420\,000$ per iteration, which is $\frac{7}{8}$ of the complexity of our design based on the nonsystematic IRA code.

Finally, the total number of loop operations in our code designs using 1024-state TCQ and the nonsystematic IRA code (with $K = 120\,000$ and $N = 480\,000$) and using 1024-state TCQ_1, 2048-state TCQ_2, and the systematic IRA code (with $K = 120\,000$ and $N - K = 360\,000$) is the same at $512 \times 4 \times 960\,000$ per BCJR decoding iteration.

VIII. Conclusion

We have addressed the hard problem of near-capacity dirty-paper code designs by following a simple source–channel coding approach of picking a TCQ code first and then designing an IRA code for the TCQ-induced channel. By employing two TCQ codebooks and optimizing the relative shift between them via tuning the EXIT charts, we are able to seamlessly combine TCQ and IRA codes in the best performing dirty-paper code design.

Compared to the nested coding approach in [10] that employs 2-D VQ plus trellis shaping for source coding and systematic doping to facilitate channel code design, our source–channel coding philosophy is much simpler with standard TCQ [19] and no systematic doping, leading to better performance at lower complexity. Compared to the superposition coding approach in [2], our main contributions are the better code designs with performance matching those predicted in Table I.

Our designs are inspired by recent work [34] on the dual problem of Wyner–Ziv coding based on TCQ and low-density parity-check (LDPC) codes. It demonstrates the benefit of leveraging advances made in the dual field of distributed source coding to that of CCSI.

Finally, the code designs in this paper and those in [25] based on nested turbo codes have enabled the studies of applications of DPC to image data-hiding [35], coding for MIMO broadcast channels [30], and transmitter cooperation [22] in wireless ad hoc networks.

ACKNOWLEDGMENT

The authors gratefully thank the anonymous reviewers for their constructive comments, which helped us improve the quality of this paper.

REFERENCES

[1] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[2] A. Bennatan, D. Burshtein, G. Caire, and S. Shamai (Shitz), "Superposition coding for side-information channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 5, pp. 1872–1889, May 2006.

[3] B. Chen and G. Wornell, "Achieving performance of digital watermarking systems," in *Proc. IEEE Int. Conf. Multimedia Computer Systems*, Florence, Italy, Jun. 1999, pp. 13–18.

[4] J. Chou, S. Pradhan, and K. Ramchandran, "Turbo coded trellis-based constructions for data embedding: Channel coding with side information," in *Proc. 35th Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2001, pp. 305–309.

[5] A. S. Cohen and A. Lapidoth, "Generalized writing on dirty paper," in *Proc. IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, Jun./Jul. 2002, p. 227.

[6] J. Conway and J. Sloane*, Sphere Packings. Lattices and Groups*. New York: Springer, 1998.

[7] M. Costa, "Writing on dirty paper," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 3, pp. 439–441, May 1983.

[8] I. Cox, M. Miller, and A. McKellips, "Watermarking as communications with side information," *Proc. IEEE*, vol. 87, no. 7, pp. 1127–1141, Jul. 1999.

[9] J. Eggers, R. Bäuml, R. Tzschoppe, and B. Girod, "Scalar costa scheme for information embedding," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 1003–1019, Apr. 2003.

[10] U. Erez and S. ten Brink, "A close-to-capacity dirty paper coding scheme," *IEEE Trans. Inf. Theory*, vol. 51, no. 10, pp. 3417–3432, Oct. 2005.

[11] U. Erez, S. Shamai (Shitz), and R. Zamir, "Capacity and lattice-strategies for cancelling known interferences," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3820–3833, Nov. 2005.

[12] T. Fischer, M. Marcellin, and M. Wang, "Trellis-coded vector quantization," *IEEE Trans. Inf. Theory*, vol. 37, no. 6, pp. 1551–1566, Nov. 1991.

[13] G. D. Forney, Jr., "Trellis shaping," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 281–300, Mar. 1992.

[14] S. Gelfand and M. Pinsker, "Coding for channel with random parameters," *Probl. Contr. and Inf. Theory*, vol. 9, pp. 19–31, 1980.

[15] A. Gersho and R. Gray*, Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.

[16] H. Harashima and H. Miyakawa, "Matched-transmission technique for channels with intersymbol interference," *IEEE Trans. Commun.*, vol. COM-20, no. 8, pp. 774–780, Aug. 1972.

[17] A. Høst-Madsen, "Capacity bounds for cooperative diversity," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1522–1544, Apr. 2006.

[18] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Intl. Symp. Turbo Codes and Related Topics*, Brest, France, Sep. 2000, pp. 1–8.

[19] M. Marcellin and T. Fischer, "Trellis coded quantization of memoryless and Gaussian-Markov sources," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 82–93, Jan. 1990.

[20] F. Petitcolas, R. Anderson, and M. Kuhn, "Information hiding—A survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[21] C. Shannon, "Channels with side information at transmitter," *IBM J. Res. Devel.*, pp. 289–293, 1958.

[22] V. Stanković, A. Høst-Madsen, and Z. Xiong, "Cooperative diversity for wireless ad hoc networks: Capacity bounds and code designs," *IEEE Signal Process. Mag.*, vol. 22, pp. 37–49, Sep. 2006.

[23] Y. Sun, A. Liveris, V. Stanković, and Z. Xiong, "Near-capacity dirty-paper code design: A source-channel coding approach," in *Proc. Conf. Information Science and Systems (CISS'05)*, Baltimore, MD, Mar. 2005.

[24] Y. Sun, A. Liveris, V. Stanković, and Z. Xiong, "Near-capacity dirty-paper code designs based on TCQ and IRA codes," in *Proc. IEEE Int. Symp. Information Theory (ISIT'05)*, Adelaide, Australia, Sep. 2005, pp. 184–188.

[25] Y. Sun, M. Uppal, A. Liveris, S. Cheng, V. Stanković, and Z. Xiong, "Nested turbo codes for the Costa problem," *IEEE Trans. Commun.*, vol. 56, no. 3, pp. 388–399, Mar. 2008.

[26] D. Taubman and M. Marcellin*, JPEG2000: Image Compression Fundamentals, Standards, and Practice*. Norwell, MA: Kluwer, 2001.

[27] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[28] M. Tomlinson, "New automatic equalizer employing modulo arithmetic," *Electron. Lett.*, pp. 138–139, Mar. 1971.

[29] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 1, pp. 55–67, Jan. 1982.

[30] M. Uppal, V. Stanković, and Z. Xiong, "Code design for MIMO broadcast channels," *IEEE Trans. Commun.*, vol. 57, no. 4, pp. 986–996, Apr. 2009.

[31] H. S. Wang and N. Moayeri, "Trellis coded vector quantization," *IEEE Trans. Commun.*, vol. 40, no. 8, pp. 1273–1276, Aug. 1992.

[32] H. Weingarten, U. Steinberg, and S. Shamai (Shitz), "The capacity region of the Gaussian multiple-input multiple-output broadcast channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 3936–3964, Sep. 2006.

[33] F. Willems, "On Gaussian channels with side information at the transmitter," in *Proc. 9th Symp. Information Theory in the Benelux*, Enschede, The Netherlands, May 1988, pp. 129–135.

[34] Y. Yang, S. Cheng, Z. Xiong, and W. Zhao, "Wyner-Ziv coding based on TCQ and LDPC codes," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 376–387, Feb. 2009.

[35] Y. Yang, Y. Sun, V. Stanković, and Z. Xiong, "Image data-hiding based on capacity-approaching dirty-paper coding," in *Proc. SPIE EI'06: Security, Steganography, and Watermarking of Multimedia Contents VIII*, San Jose, CA, Jan. 2006, vol. 6072, pp. 429–439.

[36] Y. Yang, V. Stanković, Z. Xiong, and W. Zhao, "On multiterminal source code design," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 2278–2302, May 2008.

[37] W. Yu, D. Varodayan, and J. Cioffi, "Trellis and convolutional precoding for transmitter-based interference pre-subtraction," *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1220–1230, Jul. 2005.

[38] R. Zamir, S. Shamai (Shitz), and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1250–1276, Jun. 2002.

Yong Sun received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1998 and 2001, respectively, and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, TX, in 2005.

He is now a Lead R&D Engineer in Lemko Corporation, Schaumburg, IL. His research interests include source–channel coding, wireless communications/networks, and signal processing.

Yang Yang (S'03–M'09) received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2002 and the M.S. and Ph.D. degrees in electrical engineering in 2004 and 2008 from Texas A&M University, College Station, TX.

He is currently a Postdoctoral Fellow at Texas A&M University. His research interests include joint source–channel coding, multiterminal source coding, and distributed video coding.

Angelos D. Liveris (S'03–M'05) received the Diploma in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 1998 and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, TX, in 2004.

Since 2005, he has been with Microwave Networks Inc., Stafford, TX. His research interests include distributed source coding, channel coding with side information, and wireless communications.

Dr. Liveris is corecipient of the 2006 IEEE SIGNAL PROCESSING MAGAZINE Best Paper Award.

Vladimir Stanković (M'04) received the Dipl.-Ing. degree in electrical engineering from the University of Belgrade, Serbia, in 2000 and the Dr.-Ing. degree from the University of Leipzig, Leipzig, Germany, in 2003.

From June 2003 to February 2006, he was with the Department of Electrical and Computer Engineering at Texas A&M University, College Station. From 2006 to 2007, he was with Lancaster University, Lancaster, U.K., as a Lecturer. In October 2007, he joined the Department of Electronic and Electrical Engineering, at the University of Strathclyde, Glasgow, Scotland, U.K., where he is currently a Lecturer. He has published three book chapters and over 70 papers in international journals and conference proceedings. His research focuses on image/signal processing, network information theory, wireless communications, and wireless ad hoc/sensor networks.

Dr. Stanković serves as Associate Editor of IEEE COMMUNICATIONS LETTERS.

Zixiang Xiong (S'91-M'96-SM'02–F'07) received the Ph.D. degree in electrical engineering in 1996 from the University of Illinois at Urbana-Champaign, Urabana.

From 1995 to 1997, he was with Princeton University, Princeton, NJ, first as a visiting student, then as a Research Associate. From 1997 to 1999, he was with the University of Hawaii, Honolulu. Since 1999, he has been with the Department of Electrical and Computer Engineering at Texas A&M University, College Station, TX, where he is a Professor. He spent the summers of 1998 and 1999 at Microsoft Research, Redmond, WA. His research interests are network information theory, code designs and applications, networked multimedia, and genomic signal processing.

Dr. Xiong received a National Science Foundation Career Award in 1999, an Army Research Office Young Investigator Award in 2000, and an Office of Naval Research Young Investigator Award in 2001. He also received the 2006 IEEE SIGNAL PROCESSING MAGAZINE best paper award. He served as associate editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (1999–2005), the IEEE TRANSACTIONS ON IMAGE PROCESSING (2002–2005), and the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2002–2006). He is currently an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS (Part B). He was the Publications Chair of the 2007 International Conference on Acoustics, Speech, and Signal Processing (ICASSP07) and the technical program committee Co-Chair of the 2007 Information Theory Workshop (ITW07).