

Research Article

Lossy Joint Source-Channel Coding Using Raptor Codes

O. Y. Bursalioglu,¹ M. Fresia,² G. Caire,¹ and H. V. Poor²

¹Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA

²Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA

Correspondence should be addressed to G. Caire, caire@usc.edu

Received 1 May 2008; Accepted 17 June 2008

Recommended by Massimiliano Laddomada

The straightforward application of Shannon's separation principle may entail a significant suboptimality in practical systems with limited coding delay and complexity. This is particularly evident when the lossy source code is based on entropy-coded quantization. In fact, it is well known that entropy coding is not robust to residual channel errors. In this paper, a joint source-channel coding scheme is advocated that combines the advantages and simplicity of entropy-coded quantization with the robustness of linear codes. The idea is to combine entropy coding and channel coding into a single linear encoding stage. If the channel is symmetric, the scheme can asymptotically achieve the optimal rate-distortion limit. However, its advantages are more clearly evident under finite coding delay and complexity. The sequence of quantization indices is decomposed into bitplanes, and each bitplane is independently mapped onto a sequence of channel coded symbols. The coding rate of each bitplane is chosen according to the bitplane conditional entropy rate. The use of systematic raptor encoders is proposed, in order to obtain a continuum of coding rates with a single basic encoding algorithm. Simulations show that the proposed scheme can outperform the separated baseline scheme for finite coding length and comparable complexity and, as expected, it is much more robust to channel errors in the case of channel capacity mismatch.

Copyright © 2008 O. Y. Bursalioglu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

A stationary ergodic source can be transmitted over an information-stable channel with end-to-end average distortion D with *bandwidth expansion factor* b not lower than $R(D)/C$ channel symbols per source sample, where $R(D)$ is the source rate distortion function and C is the channel capacity. Shannon's source-channel *separation principle* [1] ensures that this optimal performance can be approached by independently designing the source coding and the channel coding schemes. The bandwidth expansion factor b is defined as the number of channel symbols per source symbol. If a block of K source symbols is transmitted through the channel in N channel uses, then $b = N/K$. This provides a definite architectural advantage in practical systems, where typically (lossy) source coding is implemented at the application layer, while channel coding is designed and optimized for the physical layer.

On the other hand, this separated source-channel coding (SSCC) approach may incur substantial suboptimality, due to the nonideal behavior of finite length, finite complexity, source and channel codes. In fact, source codes designed

without taking into account the presence of channel decoding errors are typically very fragile and this might impose unnecessarily restrictive constraints on the performance of channel coding. In such cases, joint source-channel coding (JSCC) may lead to performance improvement (i.e., a better (b, D) operating point) for the same level of complexity.

Most practical lossy source coding schemes for natural sources (e.g., images, audio, video) are based on the idea of *transform coding* [2]. Source blocks are projected onto a suitable basis by a linear transformation, such that the source is well described by only a small number of significant transform coefficients. Then, the coefficients are scalar-quantized, and finally the resulting sequence of quantization indices are entropy coded. The theoretical foundation of this approach relies on the universality of entropy-coded quantization, and dates back to the work of Ziv [3]. In general, the linear transform is adapted to the given class of sources (e.g., wavelet transforms for images [4]). The statistics of the quantization indices is not known a priori. However, the memory structure of the underlying discrete source is fixed and it is typically described as a finite-memory tree-source (e.g., the context structure of JPEG2000 [5, 6]). Then,

data compression is obtained by using an adaptive entropy coding scheme that estimates the transition probabilities of the source statistical model. For example, arithmetic coding [7] with Krichevsky-Trofimov (KT) sequential probability estimation is a common choice [8].

For the sake of simplicity, this paper treats only independent and identically distributed (i.i.d.) sources with known statistics, that is, it neither deals with the transform coding aspect, nor with the universal implementation of entropy coding. However, our results can be generalized along the lines of what done in [5, 6]. Even in the nonuniversal case, classical lossless compression is *catastrophic*: a small Hamming distortion (number of bits in error) in the entropy-coded sequence is mapped into a large distortion in the reconstructed source sequence. This imposes a very strict target error probability on the channel coding stage, thus involving both complex channel coding and operating points that may be quite far from the theoretical limits. This is even more evident in applications where the coding delay is limited, thus preventing the use of very large block lengths.

It was shown in [9] that fixed-to-fixed length data compression of a discrete source with linear codes is asymptotically optimal, in the sense that compression up to the source entropy rate can be achieved. This is strongly related to transmission using the same linear code on a discrete additive noise channel where the noise has the same statistics as the discrete source. This analogy can be exploited in order to design a JSCC scheme. We wish to maintain the simplicity of the transform coding approach while improving the robustness of the scheme. The rationale behind the proposed design is the following: since linear codes achieve the entropy rate of discrete sources and the capacity of symmetric channels, we can combine the entropy coding stage and the coding stage into a single linear encoding stage. The advantage of this approach is that the design of noncatastrophic linear encoders is very well understood. Therefore, the proposed scheme can approach the optimal separation limit for large block length, while achieving better robustness to channel errors at finite decoding delay and complexity.

In [5], this JSCC approach was applied to the transmission of JPEG2000-like encoded images (in the sense that the wavelet transform, the quantization scheme and the tree-source memory structure were borrowed from JPEG2000), by using a family of progressively punctured turbo codes to map directly the redundant quantization bits into channel symbols. As stated above, here we focus on simpler i.i.d. sources with perfectly known statistics (i.e., the nonuniversal case) and investigate in greater detail the performance analysis and the comparison with the baseline SSCC approach. In this work, we use raptor codes [10] in order to map the redundant quantization bits into channel-coded symbols.

Our scheme works as follows. A source block of length K is quantized symbol by symbol. The sequence of quantization indices, represented as binary vectors, are partitioned into bitplanes. The bitplanes are separately encoded into channel symbols by a bank of binary raptor encoders. Each bitplane is encoded at a rate that depends on its conditional

entropy rate given the bitplanes previously encoded. At the decoder, the bitplanes are decoded in sequence using a multistage decoder, where in each stage we use a belief propagation (BP) iterative decoder that takes into account both the already decoded bits from previous planes, and the a priori statistics of the current bitplane as well as the received channel output.

Raptor codes are a particularly useful class of rateless codes. The advantage of using a rateless code is clear: with a single basic encoding machine we can generate a continuum of coding rates. Therefore, the scheme can adapt naturally to the entropy rate of the source and to the capacity of the channel. Although we do not pursue the universal setting in this work, we notice here that the proposed architecture allows a very fine rate matching between the (unknown a priori) source entropy and the channel capacity without resorting to a library of progressively punctured codes as is done in [5].

We express the performance of a source-channel coding scheme in terms of its peak signal-to-noise ratio (PSNR), expressed in dB, defined as

$$\text{PSNR} \triangleq -10 \log_{10}(D). \quad (1)$$

In particular, we will focus on a standard Gaussian i.i.d. source $S \sim \mathcal{N}(0, 1)$ and on the mean-squared distortion $D = \mathbb{E}[|S - \hat{S}|^2]$. In this case, the distortion-rate function is $D = 2^{-2R}$. At the Shannon separation limit, that is, letting $R = bC$, we have

$$\text{PSNR}_{\text{Shannon}} = 20 \log_{10}(2) bC = (6 \text{ dB}) \times bC. \quad (2)$$

Our aim is to design a family of practical schemes that operate close to the curve $\text{PSNR}_{\text{Shannon}}$ versus b . Notice that we do not pursue here the design of embedded schemes, that is, of single coding schemes that achieve multiple (PSNR, b) points. Nevertheless, the bitplane layered structure of the proposed encoder and the proposed multistage decoder lend themselves quite naturally to the design of embedded JSCC schemes. We leave this aspect for future work and comment on it further in the concluding section.

The rest of this paper is organized as follows. In Section 2, we review the limits of scalar entropy-coded quantization and define the target “operational Shannon limit” of our scheme. In Section 3 we present a comprehensive analysis of the baseline SSCC scheme which represents our term of comparison. Section 4 presents the details of the proposed scheme, its analysis and an algorithm for progressive incremental redundancy in order to optimize the coding rates at each bitplane. Section 5 presents some additional numerical comparisons between the baseline SSCC and the JSCC schemes, and in Section 5 we present some concluding remarks. Raptor codes, BP decoding, EXIT chart analysis and some ancillary results are presented in the appendices.

2. ENTROPY-CODED SCALAR QUANTIZATION

A source sequence of length K , $\mathbf{s} \in \mathbb{R}^K$, is quantized by applying componentwise the scalar quantizer $\mathcal{Q}_B : \mathbb{R} \rightarrow \mathbb{F}_2^{B+1}$,

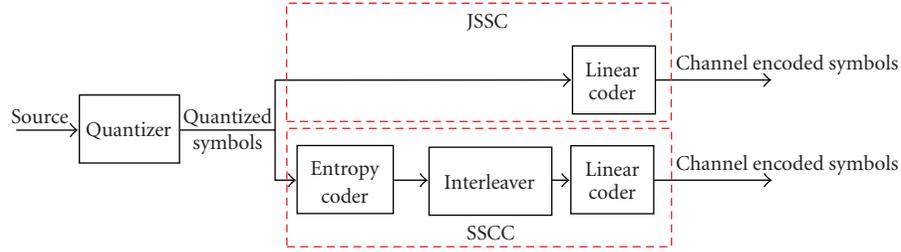


FIGURE 1: Conceptual block diagram of the conventional SSCC and the proposed JSCC schemes. The two schemes coincide but for the fact that the concatenation of entropy coding and channel coding (SSCC) is replaced by a single linear encoding block (JSCC).

where B bits are used to represent the magnitude and one bit represents the sign. Let $\mathbf{u} = \mathcal{Q}_B(\mathbf{s})$ denote the sequence of quantization indices and let $u_{p,k} : p = 0, \dots, B$ denote the bits forming the k th index. The sequence \mathbf{u} can be thought as a $(B + 1) \times K$ binary array, where each row is called a “bitplane.” Without loss of generality, we associate the 0th row with the sign bit and the rows from 1 to B with the magnitude bits, with the convention that the first bitplane is the least significant and the B th bitplane is the most significant.

As anticipated in the Introduction, we fix the quantizer and compare the performance of an SSCC approach based on the concatenation of a conventional entropy coding stage with a conventional channel code, with the performance of a JSCC that merges the two operations into a single linear encoding map. Therefore, in the absence of channel residual errors, both schemes achieve the same minimum distortion due to the quantizer, denoted by $D_{\mathcal{Q}}(B)$. Letting $H_B(U)$ denote the entropy rate of \mathbf{u} , measured in bit per quantization index, we have that the point

$$\left(b = \frac{H_B(U)}{C}, \text{PSNR} = -10 \log_{10}(D_{\mathcal{Q}}(B)) \right) \quad (3)$$

is the best achievable point for any scheme based on the fixed quantizer \mathcal{Q}_B . Following [2], we refer to this point as the “operational Shannon limit” for schemes with fixed quantizers.

We consider uniform scalar quantizers where the interval size is chosen in order to minimize the mean-squared distortion of the Gaussian unit variance i.i.d. source, for a fixed number 2^{B+1} of intervals. In [3], Ziv showed that a coding scheme formed by a scalar uniform quantizer followed by entropy coding yields a rate penalty of no more than 0.754 bits per sample with respect to the $R(D)$ limit. Thus, constraining the quantizer to be a uniform scalar quantizer should cost no more than a $0.754/C$ channel symbols per source symbols.

In Figure 2 we compare the PSNR versus b curves for the Shannon limit, the Ziv bound and the operational Shannon limit for a family of optimized uniform scalar quantizers with $B = 1, 2, 3, 4, 5$, and 6, and for channel capacity $C = 0.5$. All results in this paper make use of this family of quantizers.

3. ANALYSIS OF THE BASELINE SSCC SCHEME

In this section, we study the performance of nonideal SSCC. First, we consider the performance degradation due to

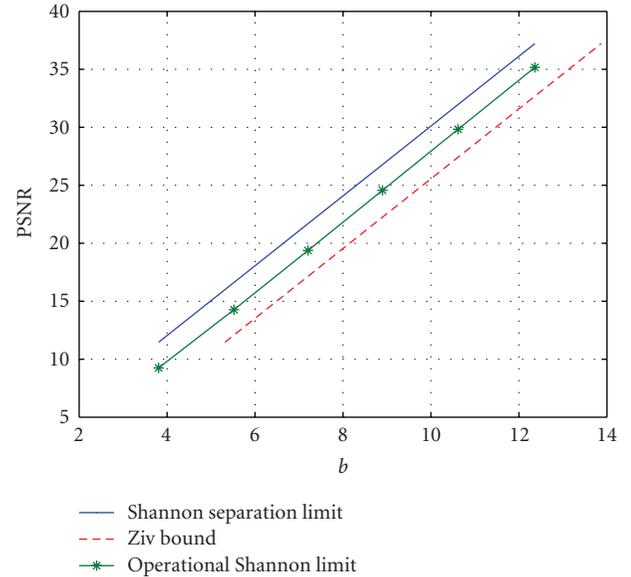


FIGURE 2: PSNR versus b for the Shannon Limit, the Ziv bound and the operational Shannon limit for the considered family of scalar quantizers for $B = 1, \dots, 6$ and for channel capacity $C = 0.5$.

nonideal source and channel codes that operate at source coding rate $R_s = R(D) + \delta_s$ and channel coding rate $R_c = C - \delta_c$, respectively, where δ_s and δ_c are positive rate gaps. This analysis assumes no errors at the output of the channel decoder.

Then, we introduce the channel decoding error probability, and obtain a distortion upper bound as a function of δ_c and δ_s , closely following the analysis of [11]. This analysis is based on the random coding exponent for channel codes, and essentially validates the error-free rate-gap analysis even for moderately large block length K .

Finally, we consider a very practical scheme, based on the concatenation of arithmetic entropy coding and a conventional binary raptor code. We provide a very accurate semi-analytic approximation for the achievable PSNR of this scheme and show that the achieved results follow closely the error free rate-gap analysis by matching the parameters δ_c and δ_s . We also notice that for finite block length K the practical scheme suffers from an additional performance degradation, especially visible at high resolution (large PSNR). We quantify this additional performance

degradation by looking at the finite-length versus infinite-length error performance of raptor codes.

3.1. Rate-gap analysis

Consider a separated scheme that makes use of channel coding at rate $R_c = C - \delta_c$ and source coding at rate $R_s = R(D) + \delta_s$, where $\delta_c, \delta_s > 0$ are rate gaps, and where the residual bit-error rate (BER) at the output of the channel decoder is (essentially) zero. Using $D = 2^{-2(R_s - \delta_s)}$ and $b = R_s/R_c$ we obtain

$$\text{PSNR} = (6 \text{ dB}) \times (b(C - \delta_c) - \delta_s). \quad (4)$$

We notice that the slope of the straight line characterizing PSNR versus b decreases with the channel coding gap δ_c , while the source coding rate gap involves only a horizontal shift. As a result, an SSCC whose channel coding stage achieves negligible BER works further and further away from the Shannon limit as PSNR increases (high resolution).

3.2. SSCC with codes achieving positive error exponent

In order to take into account channel decoding errors, we modify slightly the approach of [11] and obtain the achievable PSNR lower bound (we omit the details since the derivation follows trivially from [11]):

$$\text{PSNR} \geq (6 \text{ dB}) \times (b(C - \delta_c) - \delta_s) - 10 \log_{10}(1 + 2^{-KbE_r(C - \delta_c) + 1 + 2b(C - \delta_c) - 2\delta_s}), \quad (5)$$

where $E_r(R_c)$ denotes the random coding error exponent for a given coding ensemble over the considered transmission channel. Notice that $E_r(R_c) > 0$ for all $R_c < C$, and therefore, the error exponent is positive for all rate gaps $\delta_c > 0$. For values of $K, \delta_c, \delta_s, b, C$ such that $2^{-KbE_r(C - \delta_c) + 1 + 2b(C - \delta_c) - 2\delta_s} \ll 1$, (5) essentially coincides with (4).

Figure 3 compares (4) and (5) for different values of δ_c and for block length $K = 10000$ (which is the finite source block length that we will use throughout this paper), and $\delta_s = 0.3821$. This value of δ_s is chosen in order to match the rate gap attained by the quantizers (see Figure 5). In these results, we considered a binary symmetric channel (BSC) with capacity $C = 0.5$ (cross-over probability $\epsilon = 0.11$). The exponent $E_r(R_c)$ for the BSC can be found, for example, in [12]. For the parameters of Figure 3 we notice that (4) and (5) do not coincide for too small δ_c (e.g., $\delta_c = 0.01$ in the figure) while they coincide for large enough δ_c (in this case, $\delta_c \geq \delta_{c*} = 0.016$). For finite but large block lengths, as in this case, the threshold δ_{c*} is given by the minimum value of the channel coding rate gap above which the exponent $T(C, \delta_c, \delta_s, K, b) \triangleq -KbE_r(C - \delta_c) + 1 + 2b(C - \delta_c) - 2\delta_s$ becomes negative. In Figure 4, we plot $T(C, \delta_c, \delta_s, K, b)$ and $E_r(C - \delta_c)$ versus δ_c , when the parameters of Figure 3 are considered. It has been observed that for different values of b given in the range of Figure 3, δ_c is constant. Thus in Figure 4, we use $b = 7.2036$.

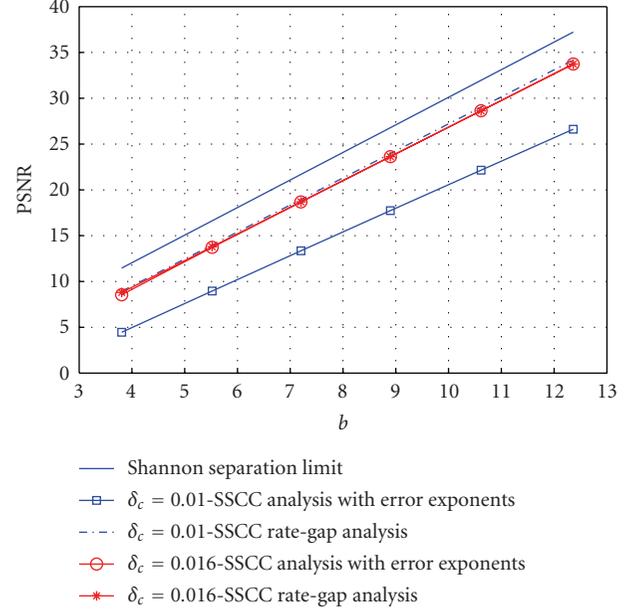


FIGURE 3: The SSCC rate-gap approximation (4) is compared with the PSNR upperbound (5) for $C = 0.5$ and $\delta_s = 0.3821$.

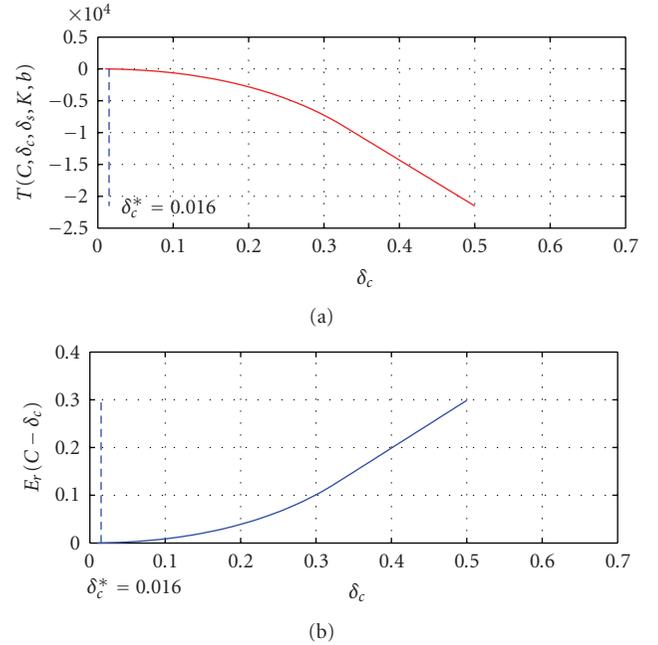


FIGURE 4: $T(C, \delta_c, \delta_s, K, b)$ and $E_r(R_c)$ versus δ_c for $C = 0.5$, $K = 10000$, $\delta_s = 0.03821$, and $b = 7.2036$.

3.3. SSCC with arithmetic coding and raptor codes

We provide an accurate approximated analysis of the performance of a practical SSCC scheme that can be regarded as our baseline scheme, since its encoding and decoding complexity is very similar to that of the JSCC scheme examined in the next section. With reference to the block diagram of Figure 1, we consider the concatenation of an

optimized uniform scalar quantizer with $B + 1$ quantization bits with an arithmetic encoder. The resulting entropy-coded bits are then channel encoded using a raptor code of suitable rate.

A sufficiently large interleaver is placed between the entropy coding and the channel coding stages, such that the bit decoding errors at the input of the arithmetic decoder can be considered to be i.i.d.. Since the arithmetic encoder has perfect knowledge of the probability distribution of the discrete source \mathbf{u} at the quantizer output, it can approach very closely the source entropy rate $H_B(U)$ even for moderate source block length K .

We approximate the performance of such a scheme by assuming that the arithmetic decoder produces random data after the first bit error at its input. Let $M = KH_B(U)$ denote the number of entropy-coded bits produced by the arithmetic encoder. These bits are channel encoded and decoded. Let m denote the position of the last correctly decoded bit before the first bit error at the arithmetic decoder input. Under the assumption of i.i.d. bit errors, m is a truncated geometric random variable with probability mass function

$$P(m = i) = (1 - P_b)^i P_b \quad (6)$$

for $i = 0, 1, \dots, M - 1$, and $P(m = M) = 1 - \sum_{i=0}^{M-1} (1 - P_b)^i P_b$, where P_b denotes the BER at the output of the channel decoder. We approximate the number of correctly decoded quantization indices by $m/H_B(U)$ (neglecting integer effects). After the first bit error, the arithmetic decoder produces random symbols distributed as the quantization indices (i.e., according to the given discrete-source probability distribution) but essentially statistically independent of the source sequence. Therefore, the average distortion in this case is given by

$$\sigma^2 = \mathbb{E}[(S - \tilde{S})^2] = \mathbb{E}[S^2] + \mathbb{E}[\tilde{S}^2] = 1 + \sigma_{\tilde{S}}^2, \quad (7)$$

where \tilde{S} denotes a random variable distributed as the quantizer reconstruction points, and $\sigma_{\tilde{S}}^2$ denotes its variance. On the other hand, before the first bit error, the system reconstructs the correct quantization points \hat{S} correctly; therefore the average distortion in this case coincides with the quantization distortion $D_{\mathcal{Q}}(B)$.

Eventually, the total average distortion of the system is approximated by

$$D \approx P(m = M)D_{\mathcal{Q}}(B) + \sum_{i=0}^{M-1} (1 - P_b)^i P_b \left[D_{\mathcal{Q}}(B) \frac{i/H_B(U)}{K} + \sigma^2 \frac{K - i/H_B(U)}{K} \right]. \quad (8)$$

The approximate analysis requires the evaluation of the residual BER at the channel decoder output. This can be obtained by simulation of the stand-alone raptor code with given finite length, or by using any suitable approximation or semi-analytic technique, such as Density Evolution or EXIT chart methods [13–17]. In particular, we make use of the EXIT chart approximation, reviewed in Appendix B.

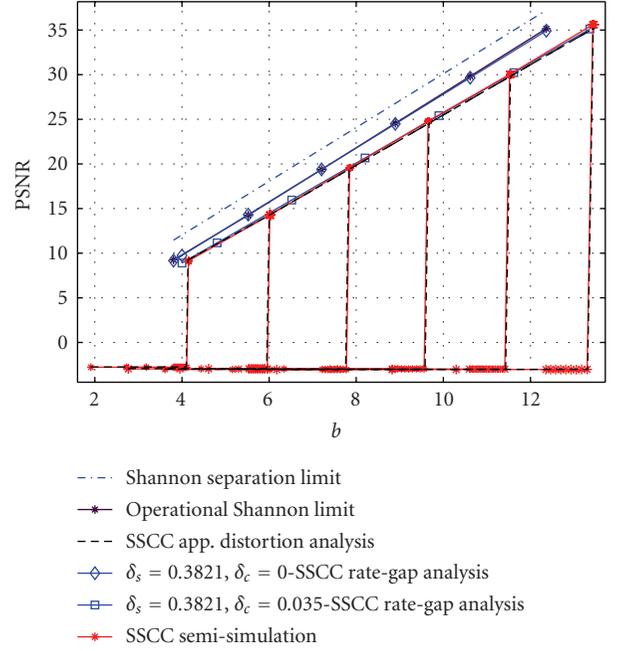


FIGURE 5: Performance of the concatenation of arithmetic coding and raptor code for infinite channel coding length, source length $K = 10000$, and a BSC with $C = 0.5$. $\delta_s = 0.3821$, and $\delta_c = 0$ are the values of rate-gaps for the operational Shannon limit (4). For the base line SSCC, these rate-gap values are empirically found to be $\delta_s = 0.3821$ and $\delta_c = 0.035$.

In Figure 5 we report the PSNR versus b (obtained by using (8)) for different values of B when $K = 10000$, a BSC with capacity $C = 0.5$ and where the raptor code output BER is approximated via the EXIT chart method. These results assume implicitly infinite channel coding block length. In order to validate the approximated distortion analysis of (8), we run simulations of the arithmetic decoder and quantization reconstruction, fed with the quantization bits corrupted by independent bit errors at a rate equal to the raptor code output BER. As seen in Figure 5, the results of the simulated arithmetic decoder match remarkably well the approximation (8), thus showing that the arithmetic decoder indeed produces approximately random data after the first bit error.

The results in Figure 5 show that for the case of very large channel coding block length the performance of the baseline SSCC scheme is remarkably close to the operational Shannon limit and therefore the scheme is hard to beat by any scheme using the same set of quantizers. However, the picture changes when we consider a finite channel coding block length. In particular, we consider independent encoding of each source block, so that the system latency is dictated by the source block length K and not by the channel coding block length. This corresponds to choosing the raptor code input bits block length equal to $M = KH_B(U)$. For the system parameters as before, the PSNR results in this case are shown in Figure 6. We notice that the finite channel coding block length yields an additional degradation that increases with PSNR.

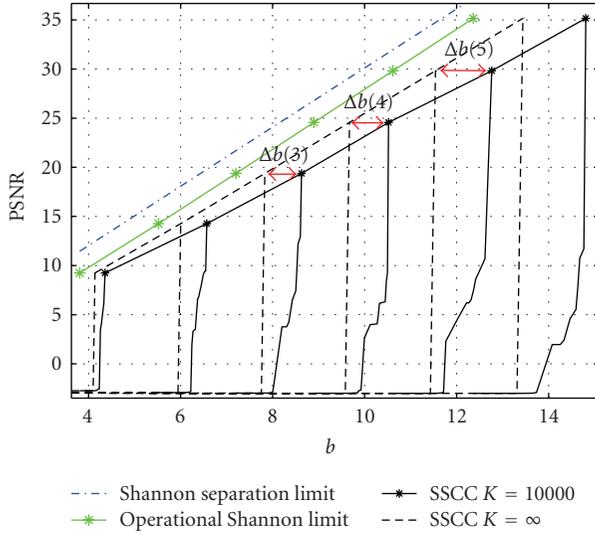


FIGURE 6: Performance of the concatenation of arithmetic coding and a raptor code for finite channel coding length, source length $K = 10000$, and a BSC with $C = 0.5$.

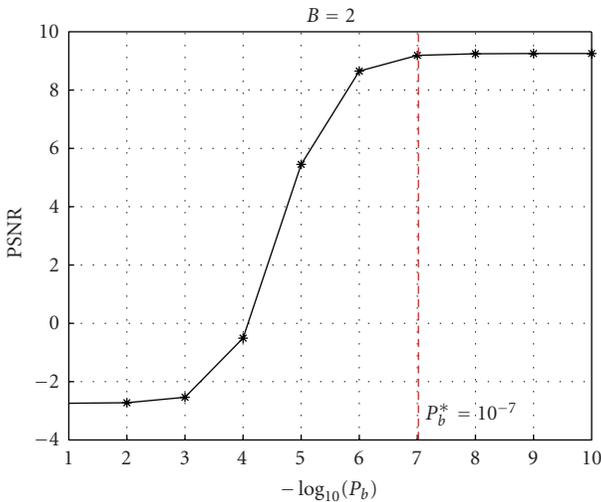


FIGURE 7: Output PSNR as a function of the channel decoding residual BER P_b .

We can explain and quantify the increasing bandwidth expansion gap $\Delta b(B)$ shown in Figure 6 as follows. Let R_{inf} and R_{fin} denote the channel coding rates needed by the raptor code to reach a small BER such that the effective distortion is virtually identical to the quantization distortion. For example, Figure 7 plots the PSNR corresponding to the distortion (8) as a function of P_b . We notice that for $P_b = 10^{-7}$ the quantization distortion (corresponding to the maximum achievable PSNR) is essentially reached. Then, Figure 8 plots the raptor code BER for the BSC with capacity $C = 0.5$, as a function of the reciprocal of the channel coding rate $1/R_c$. Notice that the raptor code is a rateless code, and therefore we can generate as many coded symbols as we like. In order to generate Figure 8 we keep the channel parameter $\epsilon = 0.11$ fixed (corresponding to $C = 0.5$) and run encoding

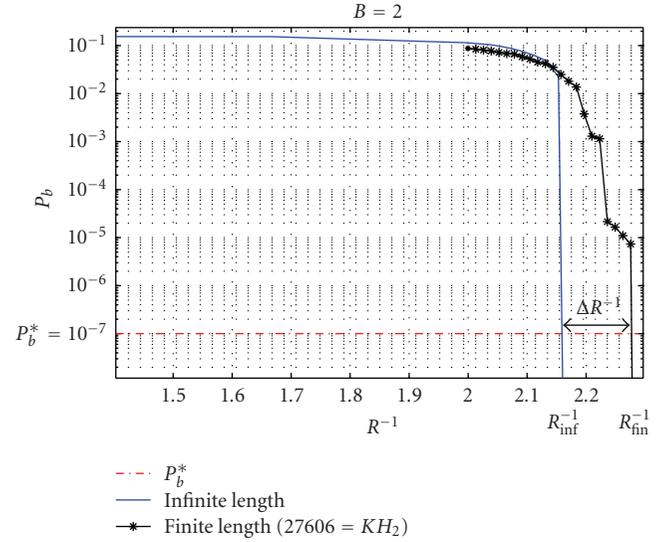


FIGURE 8: Raptor code output BER for the infinite block length case (EXIT approximation) and for the finite length case, obtained by simulation, as a function of the reciprocal of the coding rate for $C = 0.5$. The Finite length is taken to be $27606 = KH_2(U)$ which is the approximate number of bits at the output of arithmetic encoder, since the Gaussian block has entropy rate 2.7606 when quantized with \mathcal{Q}_2 .

and decoding for smaller and smaller coding rates. The infinite block length case is obtained by using the EXIT chart approximation.

Figure 8 shows that the target BER of 10^{-7} is reached at certain rates R_{inf} and R_{fin} for the cases of infinite and finite block length, respectively, and allows us to find the difference $1/R_{\text{inf}} - 1/R_{\text{fin}}$, shown in the figure.

Finally, we can quantify the bandwidth expansion gaps shown in Figure 6 by noticing that, since $b = H_B(U)/R_c$, we have

$$\Delta b(B) = H_B(U) \left(\frac{1}{R_{\text{inf}}} - \frac{1}{R_{\text{fin}}} \right). \quad (9)$$

It is clear that the gap $\Delta b(B)$ is increasing with the quantizer resolution B , and therefore with PSNR. This is a further confirmation of the fact that, in practice, it becomes more and more difficult to approach the Shannon limit as the resolution increases.

4. JOINT SOURCE-CHANNEL CODING SCHEME

In this section we describe the encoder and decoder of the proposed JSCC scheme. Then, we discuss an incremental redundancy rate allocation procedure that allows the optimization of the scheme. We hasten to say that this rate allocation procedure is run off-line, and serves to design the coding scheme for given source and channel statistics. More generally, an adaptive scheme that allocates coded bits to the bitplanes on the fly, depending on the empirical entropy rate of the source and on the capacity of the channel may be envisaged in a universal JSCC setting, where the source statistics are not known a priori and are learned instead from

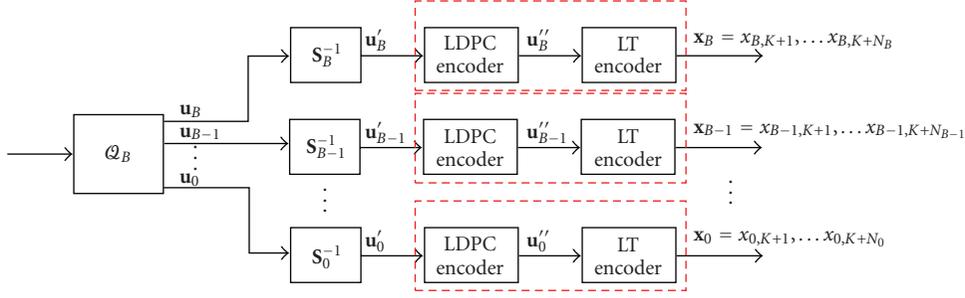


FIGURE 9: Diagram of the proposed JSCC encoder.

the source sequence itself. However, we do not pursue this approach here.

Figure 9 shows the encoder block diagram. Each bitplane (row of the binary array \mathbf{u} of quantization indices produced by the quantizer), is mapped into a sequence of coded symbols. Here we consider binary coding, and a BSC. Letting \mathbf{u}_p denote the p th row of \mathbf{u} , the corresponding block of coded symbols is given by $\mathbf{x}_p = \mathbf{u}_p \mathbf{G}_p$, where \mathbf{G}_p is a suitable encoding matrix of size $K \times N_p$. Then, the encoded blocks $\mathbf{x}_0, \dots, \mathbf{x}_B$ are transmitted in sequence over the BSC. The resulting bandwidth expansion factor is

$$b = \frac{\sum_{p=0}^B N_p}{K}. \quad (10)$$

Given the source symmetry, it is clear that the sign bit is equiprobable and has entropy $H(U_0) = 1$. Furthermore, it is independent of the magnitude bits. Hence, the target nominal rate for the encoder of the sign bit is $K/N_0 = C$. As for the p th magnitude bit, we allocate a nominal target rate equal to $K/N_p = C/H(U_p | U_{p+1}, \dots, U_B)$, where $H(U_p | U_{p+1}, \dots, U_B)$ denotes the conditional entropy rate of the p th bitplane, conditioned on the bitplanes $p+1, p+2, \dots, B$. It follows that the nominal bandwidth expansion is given by

$$\begin{aligned} b &= \frac{K/C + \sum_{p=1}^B KH(U_p | U_{p+1}, \dots, U_B)/C}{K} \\ &= \frac{1 + \sum_{p=1}^B H(U_p | U_{p+1}, \dots, U_B)}{C} \\ &= \frac{H_B(U)}{C}, \end{aligned} \quad (11)$$

which is optimal.

In order to be able to decode at these rates, we consider a multistage decoder as shown in Figure 10, that considers the bitplanes in sequence. The sign bit is independently decoded. The magnitude bits are decoded in sequence, starting from the B th plane. At each decoding stage p , the hard decisions of the already decoded planes are used by the BP decoder to compute the conditional a priori probabilities of the p th bitplane, as explained in Appendix A. Assuming that at each level p , the previous levels are correctly decoded, then the rates $K/N_p = C/H(U_p | U_{p+1}, \dots, U_B)$ are achievable.

In practice, due to the fact that the raptor codes do not achieve sufficiently low BER if their rate is too close to the

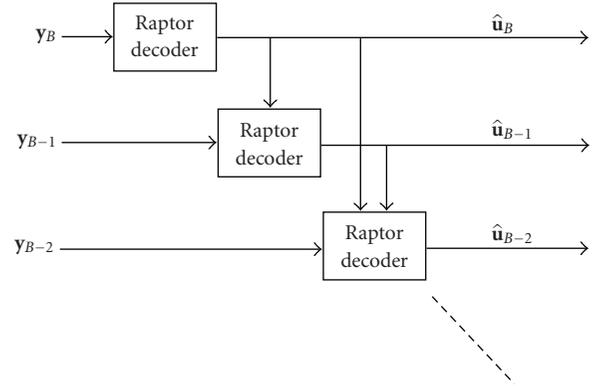


FIGURE 10: Multistage decoder for successive bitplane decoding and source reconstruction.

nominal rate limit, we must allocate the rates allowing for some gap. The rate allocation problem is made more complicated by the fact that in the multistage decoder the decoding of the different planes is not independent. In particular, if the p th plane fails with many bits in errors, then it is very likely that all the planes $p-1, p-2, \dots, 1$ will also fail, since their decoders are fed with incorrect a priori conditional probabilities. We will address the problem of rate allocation for the multistage decoder at the end of this section.

Next, let us examine in more detail how encoding of the p th plane is implemented with raptor codes [10]. Raptor codes can substantially be viewed as an extension of Luby Transform codes (LT codes) [18], since they are based on the concatenation of an outer linear code (in our case we consider low-density-parity check (LDPC) codes) with an inner LT code (see Appendix A for details). We use raptor codes in systematic form. In particular, let \mathbf{S}_p be a $K \times K$ full-rank binary matrix given by $\mathbf{S}_p = \mathbf{G}_{\text{ldpc}} \mathbf{A}_p$, where \mathbf{A}_p is a submatrix of the LT code generator matrix at encoding level p and \mathbf{G}_{ldpc} is the generator matrix of the LDPC code (see [10] for details). The encoder produces a vector of K intermediate symbols, denoted by $\mathbf{u}'_p = \mathbf{u}_p \mathbf{S}_p^{-1}$. Then, the intermediate symbols are expanded by high-rate LDPC encoding, into $\mathbf{u}''_p = \mathbf{u}'_p \mathbf{G}_{\text{ldpc}}$. Finally, the encoded symbols \mathbf{x} are obtained from \mathbf{u}'' , by applying nonsystematic rateless encoding, that is, the symbols x_1, x_2, \dots, x_{N_p} are produced in sequence, and each x_i is given as the sum of elements of \mathbf{u}'' selected at random according to the LT degree distribution Ω .

Notice that $\mathbf{u}_p = \mathbf{u}'_p \mathbf{S}_p$. Therefore, in the Tanner graph representing the code [19, 20] the nodes corresponding to source symbols \mathbf{b}_p have a degree distribution identical to that of a standard nonsystematic raptor code. Furthermore, although \mathbf{S}_p is sparse, its inverse is sufficiently dense such that the symbols \mathbf{u}'_p are close to being uniform and random i.i.d. Notice that this is essential to the scheme, since in order to drive the channel with the correct input distribution we need to send the nonsystematic symbols \mathbf{x}_p through the channel, and their distribution should be as close as possible to i.i.d. and equiprobable.

A key component in the systematic raptor code design consists of finding a suitable nonsingular $K \times K$ matrix \mathbf{S}_p , with given column weight distribution, and such that its inverse looks as much as possible like a random binary matrix. As for the LDPC code (often referred to as the “precode” in the raptor coding literature), we used a regular code with parameters (2, 100).

Let us focus now on decoding and source reconstruction. The multistage decoder of Figure 10 is based on BP at each stage p in order to approximately compute the symbol-by-symbol posterior marginal Log-Likelihood Ratios (LLRs) $\{\lambda_{p,k} : k = 1, \dots, K\}$ defined as

$$\lambda_{p,k} = \log \frac{P(u_{p,k} = 0 \mid \mathbf{y}_p, \mathbf{u}_{p+1}, \dots, \mathbf{u}_B)}{P(u_{p,k} = 1 \mid \mathbf{y}_p, \mathbf{u}_{p+1}, \dots, \mathbf{u}_B)}, \quad (12)$$

where \mathbf{y}_p denotes the channel output corresponding to the input \mathbf{x}_p , and the conditioning is with respect to the already decoded bitplanes. This is obtained by feeding the hard decisions from the planes $p + 1, \dots, B$ to the BP decoder at level p . An iterative version of the multistage decoder where soft messages in the form of a posteriori LLRs are exchanged instead of hard decisions was also considered, but it was observed that this does not provide any significant improvement and therefore was not pursued further, given its much greater complexity.

The information about the already decoded bitplanes is incorporated into the BP decoder for bitplane p in the following way. As explained in Appendix A, the BP algorithm is initialized with input messages at all the source and coded nodes in the Tanner graph of the code. The coded nodes (corresponding to the coded symbols \mathbf{x}_p), receive their input message from the corresponding channel observation. In the case of a BSC, this is given by

$$\mu_{p,i} = (-1)^{y_{p,i}} \log \frac{1 - \epsilon}{\epsilon}, \quad i = 1, \dots, N_p. \quad (13)$$

The source nodes (corresponding to the source bits \mathbf{u}_p), are associated with the input messages

$$\nu_{p,k} = \log \frac{P(u_{p,k} = 0 \mid \hat{u}_{p+1,k}, \dots, \hat{u}_{B,k})}{P(u_{p,k} = 1 \mid \hat{u}_{p+1,k}, \dots, \hat{u}_{B,k})}, \quad (14)$$

where $\hat{u}_{p+1,k}, \dots, \hat{u}_{B,k}$ are the hard decisions obtained from the previous stages.

The BP decoder at each stage runs for a given desired number of iterations, and eventually outputs both hard decisions to be passed to the next stage and soft outputs in the

form of the posterior LLRs given by (12). Once all bitplanes have been decoded, the source is reconstructed as follows. Consider without loss of generality the inverse quantization mapping

$$\mathcal{Q}_B^{-1}(u_k) = (-1)^{u_{0,k}} \frac{\Delta_B}{2} \left(\left[\sum_{p=1}^B u_{p,k} 2^p \right] + 1 \right) \quad (15)$$

that yields the mid-point of each quantization interval given the set of quantization bits.

Then, we can either consider hard reconstruction, which consists of using the hard decisions $\hat{u}_{p,k}$ in (15), or soft reconstruction, which makes use of the (approximate) posterior LLRs in order to compute the minimum-mean-square-error (MMSE) estimator of the source samples given the channel output, that is, the conditional mean estimator $\hat{s}_k = \mathbb{E}[s_k \mid \mathbf{y}]$. Treating the decoder estimated posterior LLRs as if they were the true posterior LLRs, we obtain

$$\hat{s}_k = \frac{\Delta_B}{2} \tanh \left(\frac{\lambda_{0,k}}{2} \right) \left(\left[\sum_{p=1}^B \frac{2^p}{1 + e^{\lambda_{p,k}}} \right] + 1 \right). \quad (16)$$

In Appendix A, we prove an interesting isomorphism between the BP decoder of the joint source-channel problem as described above and a related standard channel coding problem. Let us focus on a single binary independent source sequence \mathbf{u} of length K , with probabilities $p_k \triangleq P(u_k = 1)$ for $k = 1, \dots, K$. This is encoded into a binary codeword $\mathbf{x} = \mathbf{u}\mathbf{G}$, of length N , where \mathbf{G} is a $K \times N$ raptor encoder as previously described. Let us transmit \mathbf{x} through a BSC with cross-over probability ϵ , and let $\mathbf{y} = \mathbf{x} \oplus \mathbf{e}$ denote the corresponding output. The result holds for any binary input symmetric output channel, but here we focus on the BSC for simplicity of exposition. Then, the BP decoder for this problem is isomorphic to a decoder for the following related channel coding problem: consider transmission of the all-zero codeword from the systematic code with generator matrix $[\mathbf{I} \mid \mathbf{G}]$, of size $K \times (K + N)$ over a channel that for the first K components operates as

$$y_k = x_k \oplus u_k, \quad k = 1, \dots, K, \quad (17)$$

where u_k is the k th source symbol, and for the remaining N components operates as

$$y_{K+n} = x_{K+n} \oplus e_n, \quad n = 1, \dots, N. \quad (18)$$

In other words, there exists a one-to-one mapping of the messages of the BP decoder for the first problem (joint source channel) and the messages of the BP decoder for the second problem (channel only), for every edge of the decoder graph and every decoder iteration.

This means that the source-channel BP decoding can be analyzed, for example, by using the EXIT chart method, by considering the associated “virtual” channel, where the all-zero codeword from the associated systematic code is transmitted partly on a binary additive noise channel with noise realization identical to the source realization of the source-channel problem, and partly on the *same* BSC

(with the same noise realization) of the source-channel problem. We use this BP isomorphism result in order to derive a simple EXIT chart analysis of the BP decoder at each stage of the multistage decoder, under the assumption that the hard decisions from previous stages are correct.

4.1. Rate allocation algorithm

The rate allocation of each bitplane encoder is established offline by using the greedy algorithm described below. Again, we notice that we do not consider adaptive rate allocation: given the source and channel statistics, we run the greedy allocation algorithm in order to design the JSCC coding scheme.

Allocating the number of coded symbols according to the optimallimits, that is,

$$N_p = H(U_p | U_{p+1}, \dots, U_B) \frac{K}{C} \quad (19)$$

yields very bad performance even at very large block length, since it is known that raptor codes converge to very small BER at a fixed (small) gap from capacity on general binary-input symmetric output channels [14]. Therefore, we have to allow for some increment in the coded block lengths, normally referred to as “overhead” in the raptor coding literature. The problem is how to allocate a total overhead among the $B + 1$ stages. In order to do so, we propose the following greedy overhead allocation algorithm.

We initialize the lengths $N_p^{(0)}$ according to their nominal value given by (19). At each iteration of the allocation algorithm, we allocate a given number ΔN of additional coded symbols to one of the $B + 1$ codes. Let $D(N_0, \dots, N_B)$ denote the achieved average distortion of the JSCC scheme when coding lengths N_0, \dots, N_B are used and let $D^{(0)} = D(N_0^{(0)}, \dots, N_B^{(0)})$. Then, for iterations $i = 1, 2, \dots$, do the following.

- (i) For all $p = 0, \dots, B$, compute

$$D_p^{(i)} = D(N_0^{(i-1)}, \dots, N_p^{(i-1)} + \Delta N, \dots, N_B^{(i-1)}). \quad (20)$$

- (ii) Find $\hat{p} = \arg \min_{p=0, \dots, B} D_p^{(i)}$.

- (iii) Let $N_p^{(i)} \leftarrow N_p^{(i-1)}$ for all $p \neq \hat{p}$, and $N_{\hat{p}}^{(i)} \leftarrow N_{\hat{p}}^{(i-1)} + \Delta N$.

- (iv) If $|D_{\hat{p}}^{(i)} - D_Q(B)| \leq \delta$, exit. Otherwise, let $D^{(i)} \leftarrow D_{\hat{p}}^{(i)}$ and go back to 1.

The quantity $\delta > 0$ is the tolerance within which we wish to achieve the target quantization distortion.

In essence, the above algorithm allocates at each iteration a packet of ΔN -coded bits to the bitplane raptor encoder that yields the largest decrease in the overall average distortion. The distortion can be computed either by Monte Carlo simulation, or by using the EXIT chart approximation. The latter method is much faster, but cannot take into account the effect of finite block length and the error propagation between the stages of the multistage decoder.

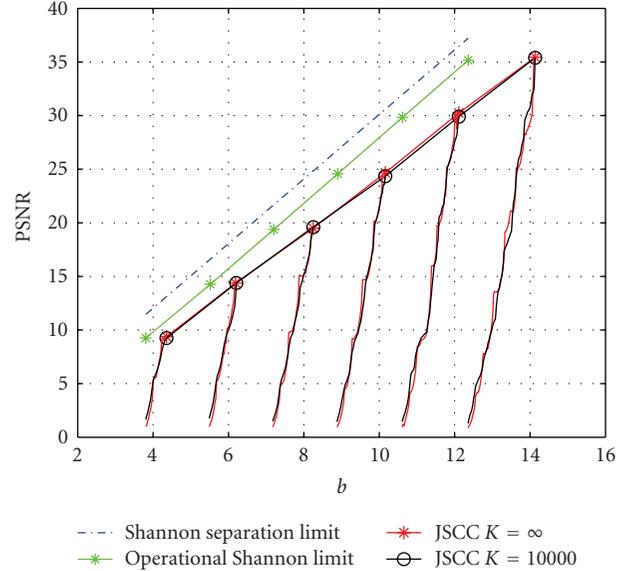


FIGURE 11: PSNR versus b comparison of JSCC scheme for finite block length simulation and infinite block length EXIT approximation for $C = 0.5$.

In Figure 11, we report the comparison between the finite length simulation and the infinite length EXIT approximation for the same setting of B ranging from 1 to 6, the BSC with capacity $C = 0.5$, and source block length $K = 10000$ used throughout the paper. As we can see, the two cases yield almost identical results. This allows us to use the infinite length EXIT approximation to estimate (with very good approximation) a suitable rate allocation among the $B + 1$ stages for the finite length case. Finally, in Figure 12, for the case of $B = 3$, we report the relative overhead $N_p^{(i)}/N_p^{(0)}$ versus $\text{PSNR}^{(i)} = -10 \log_{10} D^{(i)}$ produced by the greedy allocation algorithm. As one might expect, the greedy algorithm starts increasing the overhead of the sign bitplane and then continues from the most significant to least significant magnitude bitplanes. Eventually, each bitplane is allocated a coding length between 12% and 18% larger than the nominal length (19), in line with standard raptor coding reported results. Furthermore, notice that this scheme tends to give larger overheads to most significant bitplanes, that is, it implicitly implements unequal error protection across the layers, which is a very well-known design approach for multilevel coded modulation with multistage decoding [21].

5. NUMERICAL RESULTS

In this section, we provide both finite length and infinite length results. We considered source block length $K = 10000$ for the finite length results. In all the numerical results of this paper, we considered raptor codes with the “LT” degree distribution [14]

$$\begin{aligned} \Omega(x) = & 0.008x + 0.494x^2 + 0.166x^3 + 0.073x^4 + 0.083x^5 \\ & + 0.056x^8 + 0.037x^9 + 0.056x^{19} + 0.025x^{65} + 0.003x^{66}. \end{aligned} \quad (21)$$

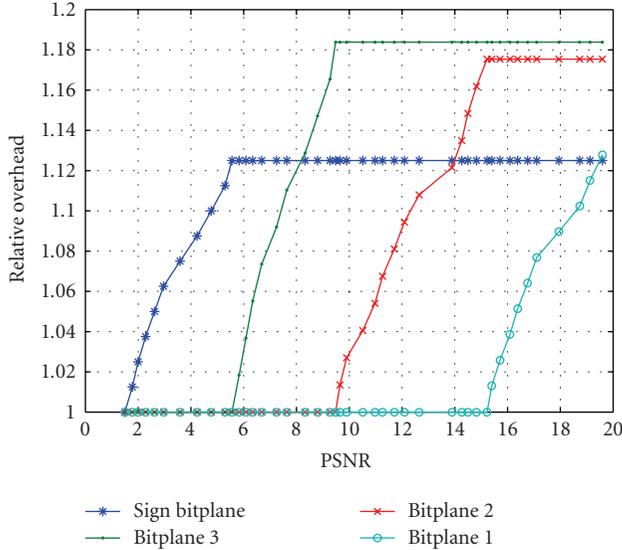


FIGURE 12: Relative overhead $N_p^{(i)}/N_p^{(0)}$ versus $\text{PSNR}^{(i)} = -10 \log_{10} D^{(i)}$ produced by the greedy allocation algorithm for the case $B = 3$. We notice that the bitplane coding overheads are incremented one at a time, in sequence.

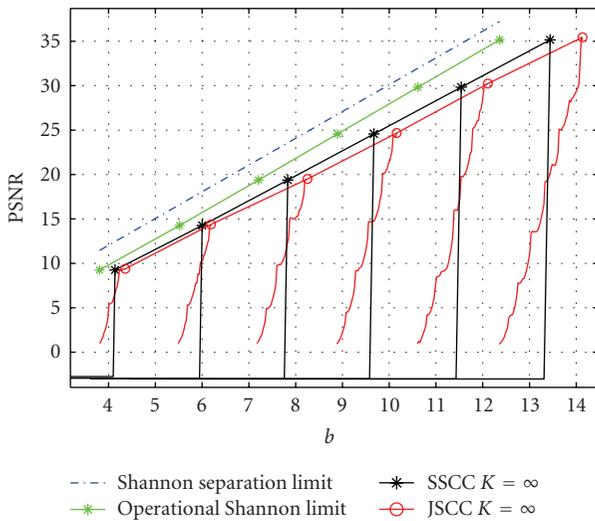


FIGURE 13: JSCC and SSCC infinite block length comparison for $B = 1, 2, 3, 4, 5, 6$ and $C = 0.5$.

As outer code we used a regular high-rate LDPC code with degrees $(2, 100)$ and rate 0.98. The source symbols are estimated after running 100 iterations of the decoding algorithm.

We would like to stress the fact that the LT and LDPC degree distribution polynomials have been chosen without considering any optimization method and that we have averaged over the ensemble of randomly generated raptor codes with the given parameters. In practice, one would carefully design an LDPC graph with good properties for the desired length K and degree distributions.

This section is subdivided in two parts. In the first part we described the results obtained by varying of the bandwidth

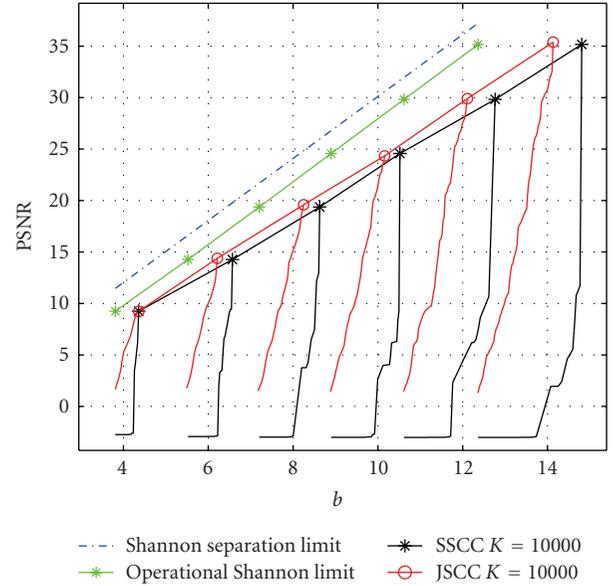


FIGURE 14: JSCC and SSCC finite block length comparison for $B = 1, 2, 3, 4, 5, 6$ and $C = 0.5$.

expansion factor, when the capacity of the BSC is fixed to $C = 0.5$, corresponding to crossover probability $\epsilon = 0.11$. The aim of this section is to compare the performance of families of SSCC and JSCC codes for different values of b , and to see how they approach the operational Shannon limit.

In the second part we examine the behavior of a *single* fixed code, designed for a nominal channel crossover probability and target PSNR, when we vary the channel crossover probability. This set of results illustrates the robustness of a given coding scheme to nonideal channel conditions.

In both subsections we provide results for infinite and finite codeword length cases. The infinite case results have been generated by using the EXIT chart approximation of Appendix B.

5.1. Approaching the operational Shannon limit

In Figure 13, we plot the performance comparison between the proposed JSCC scheme and the SSCC scheme, when infinite codeword length is considered. In this case, the SSCC scheme outperforms the proposed scheme in the sense that it reaches the quantization distortion at slightly lower values of b , for all $B = 1, \dots, 6$. The SSCC schemes show a very sharp transition (“all or nothing” behavior). In contrast, the JSCC schemes reach their quantization PSNR more gradually: as we increase the overhead, the performance improves.

The situation radically changes when we consider finite codeword length. In Figure 14 we plot the performance of JSCC and SSCC schemes for finite block length. In this case, the JSCC schemes outperform their SSCC counterpart. In particular, as we have already remarked, the JSCC performance is almost identical to that for infinite block length, while the SSCC suffers much more evidently from

the residual BER due to finite length practical codes. This also hints that the EXIT approximated analysis yields very faithful results for the JSCC scheme, while it provides optimistic results for the SSCC scheme. This can be explained by the fact that the BER performance of infinite length codes exhibits a very sharp “waterfall” threshold, beyond which the BER is zero, while for finite length the waterfall is smoother.

An important advantage of the JSCC is that the PSNR value gradually increases as b increases, while a sharp threshold effect can be seen in the case of SSCC. In [5] it was shown that, with natural sources such as images, PSNR values lower than peak value were still perceptually acceptable for the JSCC scheme, while the SSCC scheme degrades abruptly also from the perceptual viewpoint.

5.2. Robustness

In the previous set of results, we have fixed the channel capacity and the (quantized) source entropy rate and we have examined families of codes operating at different (b, PSNR) points. Now, we take a complementary view and fix the channel code while letting the channel capacity vary. This setting is relevant when a given code, designed for some nominal channel conditions, is used on a channel of variable quality, and therefore we are interested in the robustness of the performance with respect to the channel parameters. Also, this setting is more akin to the standard way of studying the performance of channel coding, where the BER is plotted as a function of the channel parameters (ϵ in the case of a BSC), for a given channel code.

In order to have a fair comparison between the two schemes, the bandwidth expansion factor (i.e., the code used) has been fixed in the following way: we keep the minimum value of b such that both schemes reach the quantization PSNR on the previous set of results (see Figure 14). In particular we keep $b = 4.3565$ and $b = 14.8079$ for $B = 1$ and $B = 6$, respectively. Since the JSCC scheme needs lower values of b to reach the quantization PSNR in both cases, we add some extra bits to reach the same values of b .

We have examined the two extreme cases of low resolution ($B = 1$) and high resolution ($B = 6$). In Figures 15 and 16 we notice that in both cases the JSCC scheme outperforms the SSCC scheme in terms of PSNR. Moreover, as expected, the PSNR of the SSCC scheme degrades sharply, while the PSNR of the JSCC scheme degrades gradually as the channel crossover probability increases. For example, considering $B = 6$, if ϵ increases from its nominal value 0.11 to a higher value 0.115 the JSCC scheme loses about 6 dB in PSNR, while the SSCC loses 24 dB. We interpret this sharp degradation as an effect of the catastrophic behavior of the entropy coding stage in SSCC, which is greatly mitigated by the linear coding stage in the proposed JSCC scheme.

6. CONCLUSIONS

Unlike most JSCC schemes presented in the literature, which are carefully targeted for specific source and channel pairs, the scheme proposed here can closely approach the rate-distortion separation limit for virtually any well-behaved

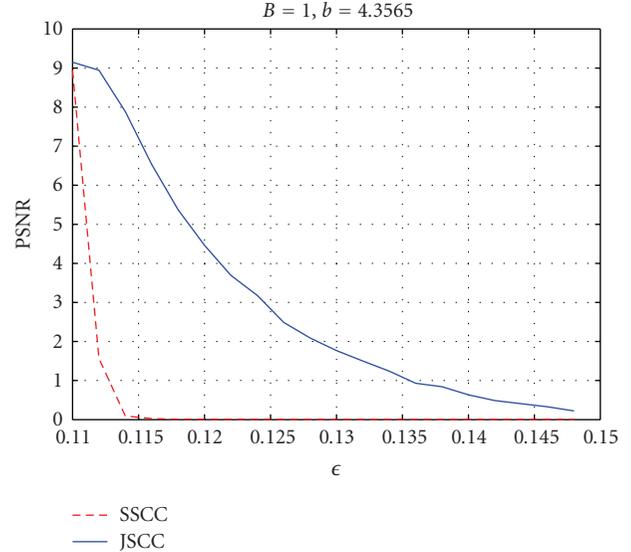


FIGURE 15: Comparison of performance degradation of JSCC and SSCC as the cross-over probability of the BSC increases for $K = 10000$ and $B = 1$.

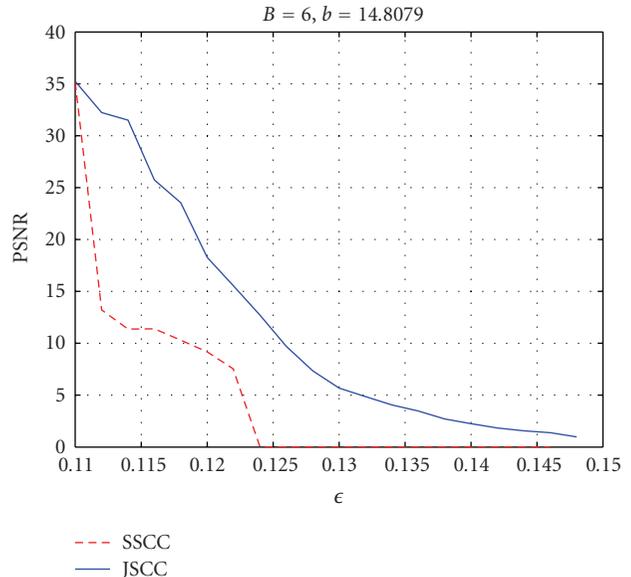


FIGURE 16: Comparison of performance degradation of JSCC and SSCC as the cross-over probability of the BSC increases for $K = 10000$ and $B = 6$.

source under quadratic distortion and any symmetric channel, owing to the universality of entropy-coded quantization and the optimality of linear codes for both data compression and channel coding. Furthermore, we have demonstrated that beyond operating close to optimal, the proposed scheme is better and more robust than a separated approach, especially in the practical case of finite block length.

We wish to conclude this paper with some considerations for future work. Following [5], the JSCC scheme can be applied to any class of sources for which efficient transform

coding has been designed. In particular, images, audio and video are natural and relevant candidates. The scheme takes advantage of the know-how and careful source statistical characterization developed in designing lossy coding standards, and preserves the structure of the transform coder. This makes it easy to introduce the JSCC scheme into practical applications, for example, by introducing a trans-coding stage at the physical layer, while preserving the network architecture and the source coding standards developed at the application layer.

Although we have not pursued this aspect here, the bit-plane layered encoding and multistage successive decoding architectures of the proposed scheme lend themselves quite naturally to a multiresolution, or “embedded,” implementation. For example, it is sufficient to use an embedded scalar quantizer in order to obtain such a scheme: bitplanes will be transmitted in sequence, and the resolution of the reconstructed source improves at each additional layer received.

A different route for future investigation involves the use of nonbinary linear codes. Also for the proposed JSCC scheme, the gap from the Shannon limit increases with the PSNR (resolution). This is due to the fact that each layer needs to be encoded with a fixed overhead, such that the overall overhead increases with the number of layers. As an alternative, we may wish to use a nonbinary raptor code operating over symbols of $B+1$ bits, and mapping directly the quantization indices over the channel symbols. The hope is that the overhead of such nonbinary codes does not depend (or at least depends in a sublinear way) on B . This may lead to better bandwidth expansion gaps at high resolution.

APPENDICES

A. RAPTOR CODES AND BP DECODING

Raptor codes [10] are a class of rateless codes designed for transmission over erasure channels with unknown capacity. They are an extension of Luby Transform codes (LT codes) [18], since they are based on the concatenation of an outer linear code (precode) with an inner LT code. To be compliant with the raptor codes terminology, let us define the *input symbols* as the symbols generated from the *source symbols* by the linear precode encoder, and *output symbols* as the symbols generated from the input symbols by the LT encoder.

Formally a raptor code is defined by the triplet $(K, \mathcal{C}, \Omega(x))$, where K is the source symbols length, \mathcal{C} is a linear encoder $\mathcal{C} : \mathbb{F}_2^K \rightarrow \mathbb{F}_2^n$ and $\Omega(x) = \sum_{j=1}^n \Omega_j x^j$ represents the generating function of the probability distribution $\Omega_1, \dots, \Omega_n$ on $\{1, \dots, n\}$ that generates the LT codewords.

The $(n, \Omega(x))$ LT code ensemble corresponds to the ensemble of $n \times N$ binary matrices, for all $N = 1, 2, \dots$, with columns randomly generated according to $\Omega(x)$, where each matrix yields an encoding mapping.

The operations to generate a generic column of an LT encoding matrix can be summarized in two steps:

- (1) sample the distribution $\{\Omega_1, \dots, \Omega_n\}$ to obtain a weight w between 1 and n ;

- (2) generate the column (v_1, \dots, v_n) uniformly at random from all $\binom{n}{w}$ binary vectors of weight w and length n ;

As shown in [14], it is possible to adapt raptor codes for transmission over memoryless symmetric channels. The decoding is performed by using the classical belief propagation algorithm (see [14] for details).

In this paper, we exploit a high rate LDPC code as the precoder, then the n input nodes can also be seen as the n bitnodes of the LDPC code.

A.1. BP decoder isomorphism

As anticipated in Section 4, there is an interesting isomorphism between the standard channel coding problem when an all zero codeword is transmitted (we refer to this as Scheme A) and the joint source-channel coding problem as defined at each stage of the multistage decoder (we refer to this as Scheme B).

Consider the following unified scheme. Let the vector $\begin{bmatrix} \mathbf{w} \\ \mathbf{z} \end{bmatrix}$ be the output block when a vector \mathbf{w} of length K is channel coded with a systematic raptor code and where \mathbf{z} has length N (i.e., the raptor code rate is equal to $(K/(K+N))$). Let us assume that the output block is transmitted over a hybrid channel such that the first K output symbols are distorted by noise vector \mathbf{u} where $p_k = P(u_k = 1)$ for $k = 1, \dots, K$ and the remaining N output symbols are distorted by the BSC channel noise vector \mathbf{e} where $P(e_k = 1) = \epsilon$, $k = 1, \dots, N$. Then, the hybrid channel is characterized by many BSCs with crossover probabilities p_1, \dots, p_K and ϵ . The channel observation block is then composed of

$$\mathbf{y} = \begin{bmatrix} \mathbf{w} \oplus \mathbf{u} \\ \mathbf{z} \oplus \mathbf{e} \end{bmatrix}. \quad (\text{A.1})$$

Notice that when $\mathbf{w} = \mathbf{0}$ then $\mathbf{z} = \mathbf{0}$, and $\mathbf{y} = \begin{bmatrix} \mathbf{u} \\ \mathbf{e} \end{bmatrix}$. In this case the unified scheme becomes Scheme A. On the other hand, when $\mathbf{w} = \mathbf{u}$, then $\mathbf{y} = \begin{bmatrix} \mathbf{0} \\ \mathbf{z} \oplus \mathbf{e} \end{bmatrix}$, and the unified scheme becomes Scheme B. Let us consider the l th iteration of the BP decoder. We use the following notation (see Figure 17):

- (i) $m_{v,o}^{(l)}$, $m_{o,v}^{(l)}$ are the messages passed from the v th input node to the o th output node and from the o th output node to the v th input node, respectively, of the LT-decoder;
- (ii) $m_{v,c}^{(l)}$, $m_{c,v}^{(l)}$ are the messages passed from the v th input node (the so called variable node in classical LDPC notations) to the c th check node and from the c th check node to the v th input node, respectively, of the LDPC decoder;
- (iii) $\delta_{\text{ldpc}}^{(l),v}$ is the message generated from the v th LDPC input node and passed to the corresponding input node of the LT-decoder;
- (iv) $\delta_{\text{lt}}^{(l),v}$ is the message generated from the v th LT input node and passed to the corresponding input node of the LDPC decoder; and

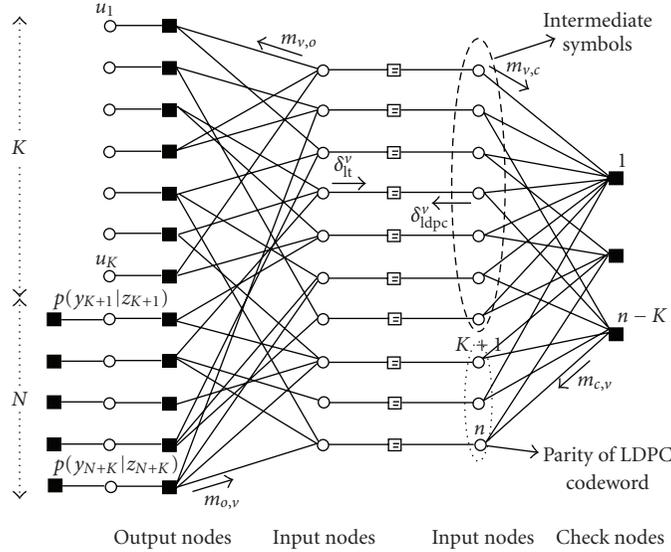


FIGURE 17: Raptor code factor graph for the application of belief propagation.

- (v) Z_o is the LLR of the o th output symbol received from noisy channel; notice for $o = 1, \dots, K$, $Z_o = (-1)^{u_o \oplus w_o} \log((1-p_o)/p_o)$ while $Z_o = (-1)^{e_o \oplus z_o} \log((1-\epsilon)/\epsilon)$ for $o = K+1, \dots, N+K$.

Using the notation above, we can define the updating rules for the LT and the LDPC decoders separately.

For the LT decoder, at the l th iteration, we have

$$\begin{aligned} \tanh\left(\frac{m_{o,v}^{(l)}}{2}\right) &= \tanh\left(\frac{(-1)^{u_o \oplus w_o} \log((1-p_o)/p_o)}{2}\right) \\ &\quad \cdot \prod_{v' \neq v} \tanh\left(\frac{m_{v',o}^{(l)}}{2}\right), \quad o = 1, \dots, K, \\ \tanh\left(\frac{m_{o,v}^{(l)}}{2}\right) &= \tanh\left(\frac{(-1)^{e_o \oplus z_o} \log((1-\epsilon)/\epsilon)}{2}\right) \\ &\quad \cdot \prod_{v' \neq v} \tanh\left(\frac{m_{v',o}^{(l)}}{2}\right), \quad o = K+1, \dots, N+K, \\ m_{v,o}^{(l+1)} &= \delta_{\text{ldpc}}^{(l),v} + \sum_{o' \neq o} m_{o',v}^{(l)}, \quad v = 1, \dots, n, \end{aligned} \quad (\text{A.2})$$

where the product is taken over all input nodes adjacent to o other than v and the summation is taken over all output nodes adjacent to v other than o . For $l = 0$, we set $m_{v,o}^{(0)} = 0$ for $v = 1, \dots, n$.

For the LDPC decoder, at the l th iteration, we have

$$\begin{aligned} m_{v,c}^{(l)} &= \begin{cases} 0 & \text{if } l = 0, \\ \delta_{\text{lt}}^{(l),v} + \sum_{c' \neq c} m_{c',v}^{(l-1)} & \text{if } l \neq 0, \end{cases} \quad v = 1, \dots, n, \quad (\text{A.3}) \\ \tanh\left(\frac{m_{c,v}^{(l)}}{2}\right) &= \prod_{v' \neq v} \tanh\left(\frac{m_{v',c}^{(l)}}{2}\right), \quad c = 1, \dots, n-K. \end{aligned} \quad (\text{A.4})$$

The messages $\delta_{\text{lt}}^{(l),v}$ and $\delta_{\text{ldpc}}^{(l),v}$ passed from the LT to the LDPC decoder and from the LDPC to the LT decoder, respectively, are defined by

$$\delta_{\text{lt}}^{(l),v} = \sum_o m_{o,v}^{(l)}, \quad v = 1, \dots, n, \quad (\text{A.5})$$

$$\delta_{\text{ldpc}}^{(l),v} = \sum_c m_{c,v}^{(l)}, \quad v = 1, \dots, n, \quad (\text{A.6})$$

where the summation is taken over all output nodes adjacent to v or overall checknodes adjacent to v .

The overall factor graph (FG) of the proposed decoding algorithm is displayed in Figure 17 for the case of JSCC $\mathbf{w} = \mathbf{u}$. We use Wiberg's notation (see [20]), that is, the FG is a bipartite graph with variable nodes (circles) and function nodes (boxes). A variable node is connected to a function node if the corresponding variable is an argument of the corresponding factor [20]. In our case, the variable nodes correspond to the input symbols of the LT code and to the input symbols of the LDPC code. The function nodes correspond to the output symbols of the LT code and to the check nodes of the LDPC code. To explicitly represent the messages passed between the two decoders at each stage, we split the graph into two parts connected to each other by "equality constraints." Finally, to distinguish between channel outputs received from the equivalent channel and channel outputs received from the noiseless channel, we explicitly represent the source symbols $\mathbf{u} = (u_1, \dots, u_K)$, and the output $\mathbf{y} = (y_{K+1}, \dots, y_{K+N})$ of the noisy channel with input \mathbf{z} . Let us also denote the input block by \mathbf{i} .

As we can see from the updating rules described above and from the factor graph, the decoder can be modeled as two independent factor graphs that exchange information between themselves after each iteration.

Theorem 1. *The magnitude of the BP messages exchanged between input and output symbols for the same Tanner graph is the same for both Schemes A and B. In particular, at BP round*

l , the relationship between the messages passed in Schemes A and B is

$$\begin{aligned} {}_B m_{v,o}^{(l)} &= (-1)^{i_v} {}_A m_{v,o}^{(l)}, \\ {}_B m_{o,v}^{(l+1)} &= (-1)^{i_v} {}_A m_{o,v}^{(l+1)}, \end{aligned} \quad (\text{A.7})$$

(where ${}_A m$ is used to denote messages for Scheme A and ${}_B m$ is used for Scheme B).

Belief propagation equations (A.2)–(A.4) can be also written in an explicit form by using a map γ from the real numbers $(-\infty, \infty)$ to $F_2 \times [0, \infty)$ defined by $\gamma(x) \triangleq (\text{sgn}(x), -\ln \tanh(|x|/2))$. Clearly γ is bijective and there exists an inverse γ^{-1} . Moreover, $\gamma(xy) = \gamma(x) + \gamma(y)$ where addition is component-wise in F_2 and in $[0, \infty)$. Another important property is as follows:

$$\gamma^{-1}\left(\sum_i \gamma((-1)^{b_i} B_i)\right) = \prod_i (-1)^{b_i} \gamma^{-1}\left(\sum_i \gamma(B_i)\right). \quad (\text{A.8})$$

Rewriting (A.2), (A.4) in terms of the γ mapping and using (A.8), we have

$$\begin{aligned} m_{o,v}^{(l)} &= \gamma^{-1}\left(\sum_{v' \neq v} \gamma(m_{v',o}^{(l)}) + \gamma((-1)^{u_o \oplus w_o} \mathcal{P}_o)\right), \\ m_{o,v}^{(l)} &= (-1)^{e_o \oplus z_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma(m_{v',o}^{(l)}) + \gamma(\xi)\right), \quad (\text{A.9}) \\ m_{c,v}^{(l)} &= \gamma^{-1}\left(\sum_{v' \neq v} \gamma(m_{v',c}^{(l-1)})\right), \end{aligned}$$

where $\mathcal{P}_o \triangleq \log((1 - p_o)/p_o)$ and $\xi \triangleq \log((1 - \epsilon)/\epsilon)$.

Similarly, we have

$$m_{o,v}^{(l)} = (-1)^{n_o \oplus z_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma(m_{v',o}^{(l)}) + \gamma(\xi)\right), \quad (\text{A.10})$$

$$m_{c,v}^{(l)} = \gamma^{-1}\left(\sum_{v' \neq v} \gamma(m_{v',c}^{(l-1)})\right). \quad (\text{A.11})$$

Proof. To prove the theorem, the BP equations for each scheme will be given explicitly and then starting with the 0th round, the relationship between the messages corresponding to different schemes will be verified. The proof follows by induction, after showing that, given the rule holds for round (l) , it also hold for round $(l+1)$.

BP for Scheme A: In this case we have

$$\begin{aligned} {}_A m_{o,v}^{(l)} &= (-1)^{u_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_A m_{v',o}^{(l)}) + \gamma(\mathcal{P}_o)\right), \\ {}_A m_{o,v}^{(l)} &= (-1)^{e_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_A m_{v',o}^{(l)}) + \gamma(\xi)\right), \quad (\text{A.12}) \\ {}_A m_{v,o}^{(l+1)} &= \sum_{o' \neq o} {}_A m_{o',v}^{(l)} + {}_A \delta_{\text{ldpc}}^{(l),v}. \end{aligned}$$

BP for Scheme B: In this case, we have

$$\begin{aligned} {}_B m_{o,v}^{(l)} &= \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_B m_{v',o}^{(l)}) + \gamma(\mathcal{P}_o)\right), \\ {}_B m_{o,v}^{(l)} &= (-1)^{e_o \oplus z_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_B m_{v',o}^{(l)}) + \gamma(\xi)\right), \quad (\text{A.13}) \\ {}_B m_{v,o}^{(l+1)} &= \sum_{o' \neq o} {}_B m_{o',v}^{(l)} + {}_B \delta_{\text{ldpc}}^{(l),v}. \end{aligned}$$

Note that in the above equations, we have provided two different versions of equations for $m_{o,v}$ for both Scheme A and Scheme B for values of $1 \leq o \leq K$ and for $K+1 \leq o \leq K+N$. We call these ranges of o the first block and the second block, respectively.

By applying the BP rules at round zero, we have the following relationships between Scheme A and Scheme B:

$$\begin{aligned} {}_A m_{o,v}^{(0)} &= {}_B m_{o,v}^{(0)} = 0, \\ {}_A m_{v,o}^{(1)} &= {}_B m_{v,o}^{(1)} = 0. \end{aligned} \quad (\text{A.14})$$

Then for round zero (A.7) are satisfied.

Now let us assume that the theorem holds for the (l) th round. Then we have the following equations for Round l

$$\begin{aligned} {}_B m_{o,v}^{(l)} &= (-1)^{i_v} {}_A m_{o,v}^{(l)}, \\ {}_B m_{\tilde{o},\tilde{v}}^{(l)} &= (-1)^{i_{\tilde{v}}} {}_A m_{\tilde{o},\tilde{v}}^{(l)}, \quad (\text{A.15}) \\ {}_B m_{v,o}^{(l+1)} &= (-1)^{i_v} {}_A m_{v,o}^{(l+1)}. \end{aligned}$$

Consequently, the equations for Round $(l+1)$ can be written as follows. Letting o and \tilde{o} denote any output symbols, from the first and the second output blocks, respectively, and letting v and \tilde{v} denote any adjacent input nodes, we can write:

$$\begin{aligned} {}_A m_{o,v}^{(l+1)} &= (-1)^{u_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_A m_{v',o}^{(l+1)}) + \gamma(\mathcal{P}_o)\right), \\ {}_A m_{\tilde{o},\tilde{v}}^{(l+1)} &= (-1)^{e_{\tilde{o}}} \gamma^{-1}\left(\sum_{\tilde{v}' \neq \tilde{v}} \gamma({}_A m_{\tilde{v}',\tilde{o}}^{(l+1)}) + \gamma(\xi)\right), \quad (\text{A.16}) \\ {}_B m_{o,v}^{(l+1)} &= \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_B m_{v',o}^{(l+1)}) + \gamma(\mathcal{P}_o)\right), \end{aligned}$$

$${}_B m_{\tilde{o},\tilde{v}}^{(l+1)} = (-1)^{e_{\tilde{o}} \oplus z_{\tilde{o}}} \gamma^{-1}\left(\sum_{\tilde{v}' \neq \tilde{v}} \gamma({}_B m_{\tilde{v}',\tilde{o}}^{(l+1)}) + \gamma(\xi)\right).$$

Using the assumption, we can write

$$\begin{aligned} {}_B m_{o,v}^{(l+1)} &= \gamma^{-1}\left(\sum_{v' \neq v} \gamma((-1)^{i_{v'}} {}_A m_{v',o}^{(l+1)}) + \gamma(\mathcal{P}_o)\right), \\ {}_B m_{\tilde{o},\tilde{v}}^{(l+1)} &= (-1)^{e_{\tilde{o}} \oplus z_{\tilde{o}}} \gamma^{-1}\left(\sum_{\tilde{v}' \neq \tilde{v}} \gamma((-1)^{i_{\tilde{v}'}} {}_A m_{\tilde{v}',\tilde{o}}^{(l+1)}) + \gamma(\xi)\right). \end{aligned} \quad (\text{A.17})$$

In order to find a relationship similar to what obtained before, we need to apply (A.8). By applying (A.8), summation coefficient terms such as $(-1)^{i_{v'}}$ or $(-1)^{i_{v'}}$ can be separated from the other summands. By (A.8), it is known that the number of terms in the summation is important. For any o , denote by \mathcal{I}_o the adjacent input node set for the output node o . Then

$$\prod_{v' \in \mathcal{I}_o, v' \neq v} (-1)^{i_{v'}} = \left(\prod_{v' \in \mathcal{I}_o} (-1)^{i_{v'}} \right) (-1)^{i_v} = (-1)^{u_o} (-1)^{i_v}, \quad (\text{A.18})$$

since the \oplus summation of all of the input nodes should give the value of the corresponding output node without additive noise. Similarly, for any \tilde{o} and \tilde{v} , define $\mathcal{I}_{\tilde{o}}$ as the set of adjacent input nodes for the output node \tilde{o} . Then

$$\begin{aligned} \prod_{\tilde{v}' \in \mathcal{I}_{\tilde{o}}, \tilde{v}' \neq \tilde{v}} (-1)^{i_{\tilde{v}'}} &= \left(\prod_{\tilde{v}' \in \mathcal{I}_{\tilde{o}}} (-1)^{i_{\tilde{v}'}} \right) (-1)^{i_{\tilde{v}}} = (-1)^{z_{\tilde{o}}} (-1)^{i_{\tilde{v}}} \\ &\Rightarrow \\ {}_B m_{o,v}^{(l+1)} &= (-1)^{u_o} (-1)^{i_v} \gamma^{-1} \left(\sum_{v' \neq v} \gamma(A m_{v',o}^{(l+1)}) + \gamma(\mathcal{P}_0) \right), \\ {}_B m_{\tilde{o},\tilde{v}}^{(l+1)} &= (-1)^{z_{\tilde{o}}} (-1)^{i_{\tilde{v}}} (-1)^{e_{\tilde{o}} \oplus z_{\tilde{v}}} \gamma^{-1} \left(\sum_{\tilde{v}' \neq \tilde{v}} \gamma(A m_{\tilde{v}',\tilde{o}}^{(l+1)}) + \gamma(\xi) \right) \\ &\Rightarrow \end{aligned} \quad (\text{A.19})$$

$${}_B m_{o,v}^{(l+1)} = (-1)^{i_v} {}_A m_{o,v}^{(l+1)}, \quad (\text{A.20})$$

$${}_B m_{\tilde{o},\tilde{v}}^{(l+1)} = (-1)^{i_{\tilde{v}}} {}_A m_{\tilde{o},\tilde{v}}^{(l+1)}. \quad (\text{A.21})$$

It is worth noting that by applying the l th round hypotheses to (A.5), we obtain ${}_B \delta_{\text{ldpc}}^{(l),v} = (-1)^{i_v} {}_A \delta_{\text{ldpc}}^{(l),v}$; that is, when we only consider the LDPC iterations, Scheme A and Scheme B differ only in the signs of the channel observations. It can easily be shown that, with such an input relationship between two schemes, the messages will be also closely related as follows:

$$\begin{aligned} {}_B m_{v,c}^{(l)} &= (-1)^{i_v} {}_A m_{v,c}^{(l)}, \\ {}_B m_{c,v}^{(l)} &= (-1)^{i_v} {}_A m_{c,v}^{(l)}, \end{aligned} \quad (\text{A.22})$$

for any l .

Then by (A.6) we obtain

$${}_B \delta_{\text{ldpc}}^{(l),v} = (-1)^{i_v} {}_A \delta_{\text{ldpc}}^{(l),v}, \quad (\text{A.23})$$

so that we can write

$$\begin{aligned} {}_A m_{v,o}^{(l+2)} &= \sum_{o' \neq o} {}_A m_{o',v}^{(l+1)} + {}_A \delta_{\text{ldpc}}^{(l+1),v}, \\ {}_B m_{v,o}^{(l+2)} &= \sum_{o' \neq o} {}_B m_{o',v}^{(l+1)} + {}_B \delta_{\text{ldpc}}^{(l+1),v}. \end{aligned} \quad (\text{A.24})$$

Applying (A.23) for round $(l+1)$,

$$\begin{aligned} {}_B m_{v,o}^{(l+2)} &= \sum_{o' \neq o} (-1)^{v_b} {}_A m_{o',v}^{(l+1)} + (-1)^{i_v} {}_A \delta_{\text{ldpc}}^{(l+1),v}, \\ {}_B m_{v,o}^{(l+2)} &= (-1)^{i_v} \left(\sum_{o' \neq o} {}_A m_{o',v}^{(l+1)} + {}_A \delta_{\text{ldpc}}^{(l+1),v} \right), \\ {}_B m_{v,o}^{(l+2)} &= (-1)^{i_v} {}_A m_{v,o}^{(l+2)}. \end{aligned} \quad (\text{A.25})$$

Equations (A.20), (A.21), and (A.25) are identical to the ones assumed in (A.15) of the (l) th round. This completes the proof by induction. \square

Due to Theorem 1, the BER of the pure channel coding scheme (assuming the all zero codeword) is equal to the BER of the source bits in the JSCC scheme. Based on this result, we can obtain an EXIT chart by considering the associated channel coding problem.

B. EXIT CHART APPROXIMATION

The standard analysis tool for graph-based codes under BP iterative decoding, in the limit of infinite block length, is density evolution (DE) [22, 23]. DE is typically computationally heavy, and numerically not very well conditioned. A much simpler approximation of DE consists of the so-called EXIT chart, which corresponds to DE by imposing the restriction that message densities are of some particular form. In particular, the EXIT with Gaussian approximation (GA) assumes that at every iteration the BP message distribution is Gaussian having a particular symmetry condition, which imposes that the variance is equal to 2 times the mean [13]. At this point, densities are uniquely identified by a single parameter, and the approximate DE tracks the evolution of this single parameter across the decoding rounds.

In particular, the EXIT chart tracks the mutual information between the message on a random edge of the graph and the associated binary variable node connected to the edge. By the isomorphism proved before, we know that the JSCC scheme and the “two-channel” scheme have the same performance. For the sake of completeness, in this section we apply the EXIT chart analysis to the “two-channel” case. The resulting EXIT chart applies directly to the JSCC EXIT chart for a binary source. Finally, we briefly discuss how to apply the EXIT chart method to the multistage decoder used by our JSCC scheme. The resulting EXIT chart analysis provides very accurate approximations of the actual JSCC scheme performance, also in the finite (moderately large) block length case (see Figure 11).

For the graph induced by the raptor (LT) distribution, we define the input nodes (also called information bitnodes), the output nodes (also called coded bitnodes) and the checknodes. For LDPC codes, we define just the bitnodes and the checknodes, since any set of bitnodes that form an information set, can be taken as information bitnodes (see Figure 17).

There are different ways of scheduling for raptor decoder.

Practical schedule

Activate in parallel all input LT checknodes, then all LDPC bitnodes (corresponding to LT input nodes), then all LDPC checknodes, then back to the LDPC bitnodes. This forms a complete cycle of scheduling, which is repeated an arbitrarily large number of times. This is the scheduling that was used in our finite length simulation.

Conceptually simple schedule

Activate the LT checknodes. Then, reset the LDPC decoder and treat the messages generated by the LT checknodes as inputs for the LDPC decoder. Perform infinite iterations of the LDPC decoder. After reaching a fixed point of the LDPC decoder, take the LLRs produced for the bitnodes by the LDPC decoder at the fixed-point equilibrium and incorporate these messages as “virtual channel observations” for the input nodes of the LT code. Then, activate all LT input nodes. This provides a complete cycle of scheduling, which is repeated an arbitrarily large number of times. Our EXIT chart equations are obtained assuming this scheduling.

EXIT charts can be seen as a multidimensional dynamic system. We are interested in studying the fixed points and the trajectories of this system. As such, an EXIT chart has state variables. Proceeding to find an EXIT recursion for the conceptually simple schedule, we will denote by \mathbf{x} and \mathbf{y} the state variables of the LT EXIT chart, and by \mathbf{X} and \mathbf{Y} the corresponding state variables for the LDPC EXIT chart.

We use the following notations.

- (i) x_i denotes the mutual information between a message sent along an edge (v, o) with “left-degree” i and the symbol corresponding to the bitnode v , and \mathbf{x} the average of “ x_i ” over all edges (v, o) . Following standard parlance of LDPC codes, we refer to the degree of the bitnode connected to an edge as the left degree of that edge, and to the degree of the checknode connected to an edge as the right degree of that edge.
- (ii) y_j denotes the mutual information between a message sent along an edge (o, v) with “right-degree” j and the symbol corresponding to the bitnode v and \mathbf{y} denotes the average of y_j over all edges (o, v) .
- (iii) X_i denotes the mutual information between a message sent along an edge (v, c) with “left-degree” i and the symbol corresponding to the bitnode v , and \mathbf{X} denotes the average of X_i over all edge (v, c) .
- (iv) Y_j denotes the mutual information between a message sent along an edge (c, v) with “right-degree” j and the symbol corresponding to the bitnode v , and \mathbf{Y} denotes the average of Y_j over all edge (c, v) .
- (v) For an LDPC code, we let $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_j \rho_j x^{j-1}$ denote the generating functions of the edge-centric left- and right-degree distributions, and we let

$$\Lambda(x) = \sum_i \Lambda_i x^i = \frac{\int_0^x \lambda(u) du}{\int_0^1 \lambda(u) du} \quad (\text{B.26})$$

denote the bit-centric left-degree distribution.

- (vi) For an LT code, we let $\iota(x) = \sum_i \iota_i x^{i-1}$ denote the edge-centric degree distribution of the input nodes, $\omega(x) = \sum_j \omega_j x^{j-1}$ denote the edge-centric degree distribution of the “output nodes” or, equivalently, the edge-centric degree distribution of the checknodes. The node-centric degree distribution of the checknodes, is given by

$$\Omega(x) = \sum_i \Omega_i x^i = \frac{\int_0^x \omega(u) du}{\int_0^1 \omega(u) du}. \quad (\text{B.27})$$

- (vii) For the concatenation of the LT code with the LDPC code we also have the node-centric degree distribution of the LT input nodes. This is given by

$$\mathfrak{I}(x) = \sum_i \mathfrak{I}_i x^i = \frac{\int_0^x \iota(u) du}{\int_0^1 \iota(u) du}. \quad (\text{B.28})$$

We consider the class of EXIT functions that make use of Gaussian approximation of the BP messages. Imposing the symmetry condition and Gaussianity, the conditional distribution of each message \mathcal{L} in direction $v \rightarrow c$ is Gaussian $\sim \mathcal{N}(\mu, 2\mu)$, for some value $\mu \in \mathbb{R}_+$. Hence, letting V denote the corresponding bitnode variable, we have

$$I(V; \mathcal{L}) = 1 - \mathbb{E}[\log_2(1 + e^{-\mathcal{L}})] \triangleq J(\mu), \quad (\text{B.29})$$

where $\mathcal{L} \sim \mathcal{N}(\mu, 2\mu)$.

In BP, the message on (v, o) is the sum of all messages incoming to v on all other edges. The sum of Gaussian random variables is also Gaussian, and its mean is the sum of the means of the incoming messages. It follows that

$$\mathbf{x}_i = J((i-1)J^{-1}(\mathbf{y}) + J^{-1}(C)), \quad (\text{B.30})$$

where C is the mutual information (capacity) between the bitnode variable and the corresponding LLR at the (binary-input symmetric output) channel output. In the raptor case, the bitnodes correspond to variables that are observed through a virtual channel by the LDPC decoder. Averaging with respect to the edge-degree distribution, we have

$$\mathbf{x} = \sum_i \iota_i J((i-1)J^{-1}(\mathbf{y}) + J^{-1}(C)). \quad (\text{B.31})$$

As far as checknodes are concerned, we use the well-known quasiduality approximation and replace checknodes with bitnodes by changing mutual information into entropy (i.e., replacing \mathbf{x} by $1 - \mathbf{x}$). Then

$$\mathbf{y}_j = 1 - J((j-1)J^{-1}(1 - \mathbf{x}) + J^{-1}(1 - C)). \quad (\text{B.32})$$

Let us consider now the “two-channel” scenario induced by the JSCC isomorphism. Let K denote the number of source bits, and N denote the number of parity bits. In the corresponding LT code, we have $M = K + N$ output nodes. The first K output nodes are “observed” through a channel with capacity $1 - H$ (i.e., the channel corresponds to the source statistics), while the second N output nodes

are observed through the actual transmission channel, with capacity C .

This channel feature is taken into account by an outer expectation in the EXIT functions. Therefore, the LT EXIT chart can be written in terms of the state equations as follows:

$$\begin{aligned} \mathbf{x} &= \sum_k \sum_i \Lambda_{ki} J((i-1)J^{-1}(\mathbf{y}) + J^{-1}(\downarrow c_k)) \\ &= \sum_k \sum_i \Lambda_{ki} J((i-1)J^{-1}(\mathbf{y}) + kJ^{-1}(\mathbf{Y})), \end{aligned} \quad (\text{B.33})$$

where $K/M = \beta$ and $N/M = 1 - \beta$, and

$$\begin{aligned} \mathbf{y} &= 1 - \sum_j \omega_j [\beta J((j-1)J^{-1}(1-\mathbf{x}) + J^{-1}(H)) \\ &\quad + (1-\beta)J((j-1)J^{-1}(1-\mathbf{x}) + J^{-1}(1-C))], \end{aligned} \quad (\text{B.34})$$

where $\downarrow c_k$ is the mutual information input by the LDPC graph into the LT code graph via the node v of degrees (i, k) as explained in the following.

Equation (B.34) follows from the fact that a random edge (o, v) is connected with probability β to a source bit (i.e., to the channel with capacity $1 - H$), while with probability $1 - \beta$ to a parity bit (i.e., to the channel with capacity C).

Consider an LDPC bitnode v that coincides with an input node of the LT code. The degree of this node with respect to the LDPC graph is k , while the degree of v with respect to the LT graph is i . For a randomly generated graph, and a random choice of v , k and i are independent random variables, with joint distribution given by

$$\prod_{i,k} \mathfrak{I}_i \Lambda_k. \quad (\text{B.35})$$

The mutual information input by the LT graph into the LDPC graph via the node v of degrees (i, k) is given by

$$\uparrow c_i = J(iJ^{-1}(\mathbf{y})). \quad (\text{B.36})$$

Therefore, the LDPC EXIT chart can be written in terms of the following state equations:

$$\begin{aligned} \mathbf{X} &= \sum_k \sum_i \lambda_k \mathfrak{I}_i J((k-1)J^{-1}(\mathbf{Y}) + J^{-1}(\uparrow c_i)) \\ &= \sum_k \sum_i \lambda_k \mathfrak{I}_i J((k-1)J^{-1}(\mathbf{Y}) + iJ^{-1}(\mathbf{y})), \\ \mathbf{Y} &= 1 - \sum_\ell \rho_\ell J((\ell-1)J^{-1}(1-\mathbf{X})). \end{aligned} \quad (\text{B.37})$$

The mutual information input by the LDPC graph into the LT graph via the node v of degrees (i, k) is given by

$$\downarrow c_k = J(kJ^{-1}(\mathbf{Y})). \quad (\text{B.38})$$

Equations (B.37), (B.33), and (B.34) form the state equations of the global EXIT chart of the concatenated LT-LDPC graph, where the state variables are \mathbf{x} , \mathbf{y} , \mathbf{X} , \mathbf{Y} , while

the parameters are H , C and β , and the degree sequences ω , ι , ρ , and λ .

Finally, in order to get the reconstruction distortion we need to obtain the conditional probability density function (pdf) of the LLRs output by BP for the source bits. Under the Gaussian approximation, the LLR is Gaussian. Let μ_j denote the mean of the LLR of a source bitnode corrected to a checknode of degree j , given by

$$\mu_j = J^{-1}(1 - J(jJ^{-1}(1 - \mathbf{x}))) + J^{-1}(1 - H). \quad (\text{B.39})$$

Then, we approximate the average BER of the source bits as

$$P_b = \sum_j \Omega_j Q\left(\sqrt{\frac{\mu_j}{2}}\right). \quad (\text{B.40})$$

B.1. Multilayer EXIT chart analysis

For each bitplane, at every location, the entropy of the bit depends on the realization of the bits at the previous (more significant) bitplanes. We are then in the presence of a ‘‘time-varying’’ memoryless channel in the corresponding channel coding problem. To develop the equations for the multilayer case, we use the same idea described in the previous section, namely, an outer expectation. At the $(B - p)$ th most significant level, the previous corresponding bit locations might have 2^p different combinations, with possibly different probabilities which are denoted by $\gamma_{B-p}(m)$ for $0 \leq m \leq 2^p - 1$.

Let $H(x_{B-p}|m)$ denote the conditional entropy of a bit at the $(B - p)$ th most significant plane given that the value of the corresponding more significant bits’ combination is m .

At the $(B - p)$ th most significant level, the channel has capacity C with probability $1 - \beta$, while it has capacity $1 - H(x_{B-p}|m)$ for $0 \leq m \leq 2^p - 1$ with probability $\beta \gamma_{B-p}(m)$.

Following this approach we can modify (B.34) for the decoding of the $(B - p)$ th magnitude plane. It is worth noting that when the $(B - p)$ th biplane is considered, we assume that the sign plane and p (from B to $B - p - 1$) magnitude planes have been processed. Since it is known that the magnitude plane model does not depend on the sign plane, we take into account 2^p different realizations.

That is, we have

$$\begin{aligned} \mathbf{y} &= 1 - \sum_j \omega_j \left\{ (1 - \beta)J((j-1)J^{-1}(1-\mathbf{x}) + J^{-1}C) \right. \\ &\quad \left. + \beta \left(\sum_{m=0}^{2^p-1} \gamma_{B-p}(m) J((j-1)J^{-1}(1-\mathbf{x}) + J^{-1}(H(x_{B-p}|m))) \right) \right\}, \\ &\quad p = 1, \dots, B - 1. \end{aligned} \quad (\text{B.41})$$

We would like to underline that the sign bitplane and the most important biplane do not depend on any other bitplane, and so $\gamma_B(0) = \gamma_0(0) = 1$.

Similar to (B.41), we can update (B.40) as follows:

$$P_b = \sum_j \Omega_j Q\left(\sqrt{\frac{\mu_j}{2}}\right), \quad (\text{B.42})$$

where

$$\mu_j = J^{-1}(1 - J(jJ^{-1}(1 - x))) + \sum_{m=0}^{2^p-1} \gamma_{B-p}(m) J^{-1}(1 - H(x_{B-p}|m)). \quad (\text{B.43})$$

Note that a genie-aided scheme was assumed for the EXIT analysis, where there is no error-propagation between layers. In fact, it is not possible to take into account error propagation using the EXIT chart, since the underlying assumption is that the message exchanged at each iteration of the BP is a true LLR, that is, an LLR computed on the basis of the correct conditional probabilities. Decision errors, instead, would feed the decoder at a lower stage with “false” a priori probabilities.

As was done in the finite-length case, we will use soft reconstruction for the infinite-length case where the conditional mean estimator of the reconstruction points will be calculated using the fact that the source bit LLRs have the symmetric Gaussian distribution. From the EXIT chart, we can obtain the mean μ of the Gaussian approximation of the conditional pdf of any LLR in the graph. Hence, this can be used to compute the MMSE of the soft-reconstruction estimator (16).

ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation under Grants ANI-03-38807, CNS-06-25637 and NeTS-NOSS-07-22073 and in part by the USC Annenberg Graduate Fellowship Program.

REFERENCES

- [1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 1991.
- [2] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.
- [3] J. Ziv, “On universal quantization,” *IEEE Transactions on Information Theory*, vol. 31, no. 3, pp. 344–347, 1985.
- [4] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practices*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.
- [5] M. Fresia and G. Caire, “A practical approach to lossy joint source channel coding,” in *Proceedings of the Information Theory and Applications Workshop (ITA '07)*, San Diego, Calif, USA, February 2007.
- [6] M. Fresia and G. Caire, “A practical approach to lossy joint source-channel coding,” submitted to *IEEE Transactions on Information Theory*.
- [7] I. H. Witten, R. M. Neal, and J. G. Cleary, “Arithmetic coding for data compression,” *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, 1987.
- [8] R. E. Krichevsky and V. K. Trofimov, “The performance of universal encoding,” *IEEE Transactions on Information Theory*, vol. 27, no. 2, pp. 199–207, 1981.
- [9] G. Caire, S. Shamai, and S. Verdú, “Noiseless data compression with low-density parity-check codes,” in *Advances in Network Information Theory*, P. Gupta, G. Kramer, and A. J. van Wijngaarden, Eds., vol. 66 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 263–284, American Mathematical Society, Providence, RI, USA, 2004.
- [10] A. Shokrollahi, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
- [11] B. Hochwald and K. Zeger, “Tradeoff between source and channel coding,” *IEEE Transactions on Information Theory*, vol. 43, no. 5, pp. 1412–1424, 1997.
- [12] R. G. Gallager, *Information Theory and Reliable Communication*, John Wiley & Sons, New York, NY, USA, 1968.
- [13] S. T. Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.
- [14] O. Etesami and A. Shokrollahi, “Raptor codes on binary memoryless symmetric channels,” *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 2033–2051, 2006.
- [15] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.
- [16] A. Roumy, S. Guemghar, G. Caire, and S. Verdú, “Design methods for irregular repeat-accumulate codes,” *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1711–1727, 2004.
- [17] R. Palanki and J. S. Yedidia, “Rateless codes on noisy channels,” in *Proceedings of the Conference on Information Sciences and Systems (CISS '04)*, Princeton, NJ, USA, March 2004.
- [18] M. Luby, “LT codes,” in *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS '02)*, pp. 271–280, Vancouver, Canada, November 2002.
- [19] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.
- [20] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.
- [21] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, “Multilevel codes: theoretical concepts and practical design rules,” *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1361–1391, 1999.
- [22] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, “Analysis of low density codes and improved designs using irregular graphs,” in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC '98)*, pp. 249–258, Dallas, Tex, USA, May 1998.
- [23] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

