Compression of a Binary Source with Side Information Using Parallelly Concatenated Convolutional Codes

Zhenyu Tu, Jing Li (Tiffany) and Rick S. Blum
Electrical and Computer Engineering Department
Lehigh University, Bethlehem, PA 18015
Email: {zht3, jingli, rblum}@ece.lehigh.edu

Abstract— **This paper presents an efficient structured binning scheme for solving the noiseless distributed source coding problem with parallel concatenated convolutional codes, or turbo codes. The novelty in the proposed scheme is the introduction of a syndrome former and an inverse syndrome former to efficiently and optimally exploit an existing turbo code without the need to redesign or modify the code structure and/or decoding algorithms. Extension of the proposed approach to serially concatenated codes is also briefed and examples including conventional turbo codes and asymmetric turbo codes are given to show the efficiency and the general applicability of the approach. Simulation results reveal good performance which is close to theoretic limit.**

I. Introduction

The challenging nature of multi-user communication problems [1] has been recognized for decades and many of these problems still remain unsolved. Among them is the distributed source coding (DSC) problem, which refers to the compression of two or more physically-separated but statistically-correlated information sources, where the sources (e.g. sensors) send the (compressed) information to a central point (e.g. monitoring station) without communicating with each other. The theory and conceptual underpinnings of the noiseless DSC problem started to appear back in the seventies [2][3][4][5]. In particular, Slepian-Wolf theorem states that separate encoding (but joint decoding) does not incur a loss in capacity from joint encoding [2]. The random binning idea used in the proof of Slepian-Wolf theorem requires a structured binning implementation in practice [6]. The first constructive algebraic binning scheme was demonstrated by Wyner [1] where cosets of a linear parity check code are used as bins and a syndrome decoder is used to decode the codes. This approach was further extended to include non-syndrome decoders by Pradhan and Ramchandram [7]. Since then, several practical coding schemes have been proposed for use in DSC, including coset codes [7], lattice codes [8][6], low density parity check (LDPC) codes [9][10][11] and more recently turbo codes [12][13][14][15][16].

This material is based on research supported by the Air Force Research Laboratory under agreement No. F49620-03-1-0214, by the National Science Foundation under Grant No. CCR-0112501 and CCF-0430634, and by the Commonwealth of Pennsylvania, Department of Community and Economic Development, through the Pennsylvania Infrastructure Technology Alliance (PITA).

In this work, we propose to exploit parallel concatenated convolutional codes (PCCC), or turbo codes, for near-lossless DSC. The goal is to get close to the theoretic limit with as low complexity as possible. The reasons for using turbo codes are three-fold: (i) turbo codes are capacity-approaching channel codes, (ii) a turbo encoder is cheap to implement (thus appealing for applications like sensor networks where the computation on the transmitter side needs to be minimized), (iii) unlike random LDPC codes, the length of a turbo code is flexible (easily changed), making it possible to track and adapt to the varying correlation between sources.

Among the available turbo-DSC literature, Garcia-Frias and Zhao were the first to propose an interesting turbo scheme where two sources were separately turbo encoded and decoded in an interwoven way akin to a four-branch turbo code [12]. A similar scheme that works for asymmetric compression was independently devised by Aaron and Girod [13]. In [14], Bajcsy and Mitran proposed yet another turbo structure based on finite-state machine codes. The work of Liveris, Xiong and Georghiades is particularly worth mentioning [16]. It is the first to optimally exploit an existing turbo code, but does not make explicit use of the binning scheme. The implicit binning approach therein involves merging a principle trellis with a complementary trellis to construct a source coding trellis that contains parallel branches. If the component convolutional code has rate $1/k$, then there are 2^{k-1} parallel branches between a valid pair of states. Encoding is performed by a walk through the trellis (or the state diagram) and decoding requires a modified turbo decoder to accommodate the time-varying trellis [16]. These pioneering works [12][13][14][16] revealed the potential of the turbo-DSC solution, but *simple and universal* binning schemes to *fully exploit* an *existing* turbo code *without redesigning the code or modifying the decoder* are not entirely clear. This is where this paper fits in. Compared to the approach in [16], the proposed one provides an *explicit* algebraic binning realization for turbo codes which is simpler and more efficient since there is no need to go through code trellis during encoding and the decoding trellis is not time-variant.

We note that the binning idea first introduced in [2][1] provides a fresh and sharp tool that allows all linear channel codes (also known as error correction codes or ECC) to be fully exploited for distributed source coding. The idea is to group data sequences (i.e. virtual codewords) into bins/cosets and, by transmitting the (short) bin-indexes instead of the (long) codewords, compression is achieved. In order to (near-)losslessly recover the information at the decoder, a "geometrical uniformity" property needs to be preserved in all bins. More details will follow in Section II. Here, we wish to point out that, capitalizing the general framework of binning approach, the task left for a specific channel code is to find a practical and efficient way to bin codewords and to index bins. Whereas block codes are easily "binned", the random interleaver in a turbo code makes the task tricky. This probably explains why previous works on turbo-DSC problems have not fully exploited the binning approach. (As mentioned before, the turbo-DSC works of Liveris *et al* [16] are the only ones known to the authors that have implicitly used the binning idea.) The novelty in the proposed method is that we introduce a simple way to construct a *syndrome former* (SF) and an *inverse syndrome former* (ISF) at the encoder and decoder, respectively, to efficiently bin codewords without the need to explicitly write out the parity check matrix (near-impossible task) or redesign/modify the code structure or the decoding algorithms. The approach is simple, optimal, and generally applicable to all turbo codes, including asymmetric turbo codes [17]. Simulations on both conventional turbo codes and asymmetric turbo codes with fairly large block sizes reveal a compression rate that is extremely close to the theoretic limit.

The rest of the paper is organized as follows. Section II formulates the DSC problem and presents the generic binning scheme for compression with side information (SI). Section III discusses in detail the proposed approach, starting with convolutional codes, and moving on to parallel turbo codes and finally serial turbo codes. Section IV presents simulation results and Section V concludes the paper.

II. Background

A. System Model of the DSC Problem

Let us first formulate the setting for discussion. Consider two correlated binary memoryless sources X and Y encoded by separated encoders and decoded by a joint decoder. The achievable rate region is given by the Slepian-Wolf boundary [2]:

$$R_1 \geq H(X|Y) \quad (1)$$

$$R_2 \geq H(Y|X) \quad (2)$$

$$R_1 + R_2 \geq H(X,Y) \quad (3)$$

where R_1 and R_2 are the compression rates of X and Y, respectively. A typical illustration is given in Fig. 1.

For most cases of practical interest, zero-error DSC is possible only asymptotically [18]. For discrete symmetric sources, however, corner points on the Slepian-Wolf boundary can be achieved by considering one source (e.g. Y) as the side information and compressing the other (i.e. X) to its conditional entropy ($H(X|Y)$). The line connecting the corner points can then be achieved through time-sharing.

B. The Generic Binning Scheme with Binary Linear Codes

This subsection outlines the generic binning approach to achieve the corner point of A or B, which is essentially the problem of compression with side information. The correlation between X and Y is modeled by a virtual binary symmetric channel (BSC) as $X = Y \oplus Z$, where Z is a Bernoulli process with $Prob(Z = 1) = p$ and is independent of Y. Fig. 2 shows the system diagram.

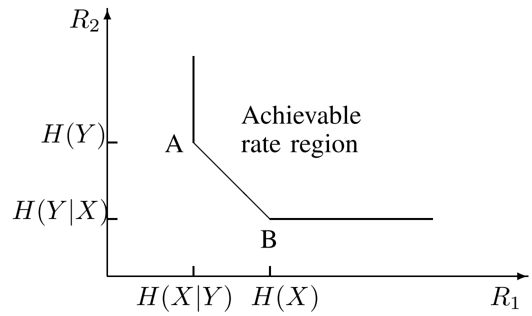

Fig. 1. Rate region for noiseless DSC

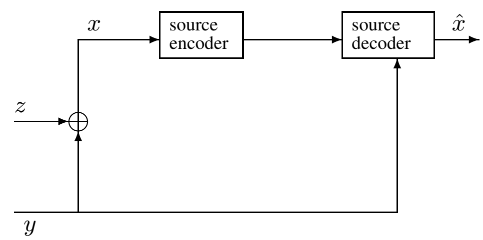

Fig. 2. Source coding with side information at decoder

The binning/coset/syndrome approach of DSC uses a (powerful) linear error correction code to construct the bins or the non-overlapping cosets of the codeword/sequence space. The assignment of the bin-indexes or the syndromes are not unique, except for the coset of valid codewords whose coset syndrome is always $\mathbf{0}$. Let $\mathbf{c}(\mathbf{s})$ denote a codeword $\mathbf{c}$ having syndrome $\mathbf{s}$. The encoder and decoder proceed as follows:

- **Encoder:** compress source sequence $\mathbf{x} = \mathbf{c_1}(\mathbf{s})$ to its syndrome $\mathbf{s}$.
- **Noiseless channel:** send $\mathbf{s}$ to the decoder without error.
- **Decoder:** Choose an arbitrary sequence $\mathbf{c_2}(\mathbf{s})$ from the coset of $\mathbf{s}$, subtract it from side information $\mathbf{y}$, treat the resulting sequence as a noisy codeword and perform ECC decoding. Notice

$$\begin{aligned} \mathbf{y} \oplus \mathbf{c_2}(\mathbf{s}) &= \mathbf{x} \oplus \mathbf{c_2}(\mathbf{s}) \oplus \mathbf{z} && (4) \\ &= \mathbf{c_1}(\mathbf{s}) \oplus \mathbf{c_2}(\mathbf{s}) \oplus \mathbf{z} && (5) \\ &= \mathbf{c_3}(\mathbf{0}) \oplus \mathbf{z}; && (6) \end{aligned}$$

  i.e., $\mathbf{y} \oplus \mathbf{c_2}(\mathbf{s})$ can be equivalently viewed as a noise-corrupted version of $\mathbf{c_3}(\mathbf{0})$ (transmitted through a virtual BSC). Hence, if the ECC is powerful enough, codeword $\mathbf{c_3}(\mathbf{0})$ can be recovered with vanishing error probability. Adding $\mathbf{c_2}(\mathbf{s})$ back to $\hat{\mathbf{c_3}}(\mathbf{0})$ then yields the estimate for the original source sequence $\mathbf{x}$ (or $\mathbf{c_1}(\mathbf{s})$), denoted by $\hat{\mathbf{x}}$. Notice $Prob(\hat{\mathbf{x}} \neq \mathbf{x}) = Prob(\hat{\mathbf{c_3}}(\mathbf{0}) \neq \mathbf{c_3}(\mathbf{0}))$. It follows that $\mathbf{x}$ is also recovered with vanishing distortion.

Notice that the capacity of the virtual BSC is $1 - H(Z)$. A capacity-achieving code has rate $R \approx 1 - H(Z)$, and the syndrome sequence has rate $(1 - R)$ which achieves the theoretic limit $H(X|Y) = H(Z)$. This shows the optimality of the binning approach. The immediate implication is that a capacity-approaching channel code can achieve capacity-approaching source coding or, equivalently, to find a good distributed source coding scheme is to find a good channel coding scheme on the virtual channel and to implement an efficient binning scheme for it.

III. Structured Binning with Parallel and Serial Turbo Codes

Following the algebraic binning scheme illustrated above, we discuss in detail below how to efficiently and optimally exploit it with turbo codes. We start with convolutional codes which are the component codes of a turbo code, followed by a detailed discussion on (parallel) turbo codes. Extensions to serial turbo codes, or serially concatenated convolutional codes (SCCC), is also presented, but in brevity.

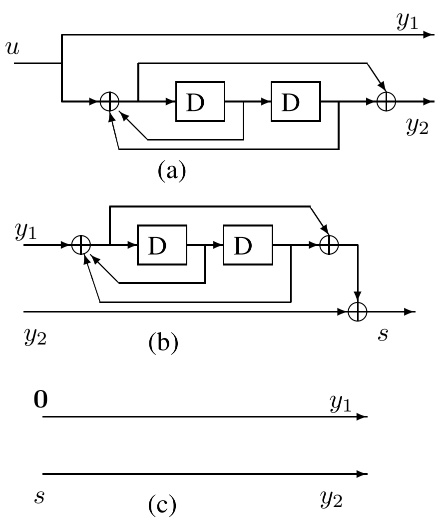

Fig. 3. (a) A rate 1/2 systematic recursive convolutional encoder. (b) Syndrome Former. (c) Inverse Syndrome Former.

A. A Component Convolutional Code

In his 1992 paper on trellis shaping [19], Forney described a simple way to construct syndrome formers and inverse syndrome formers for convolutional codes. For a rate k/n binary linear convolutional code C with $k \times n$ generator matrix G[1], it is shown that the SF can be implemented using an $n/(n-k)$ linear sequential circuit specified by an $n \times (n-k)$ transfer matrix H^T with rank $(n-k)$ such that

$$GH^T = \mathbf{0}. \tag{7}$$

This constraint makes sure that all valid codewords are associated with the all-zero syndrome $\mathbf{0}$ and that length-n codewords/sequences have the same syndrome if and only if they belong to the same coset. The inverse syndrome former, $(H^{-1})^T$, takes the left inverse of the SF and generates a coset representative sequence from a syndrome sequence $\mathbf{s}$, i.e.

$$(H^{-1})^T H^T = I. \tag{8}$$

It should be noted that the SF-ISF pairs are not unique for a given code. Any linear sequential circuit having the required number of inputs and outputs and meeting the constraints of (7) and (8) should work, but complexity could vary. As an example, consider a rate 1/2 recursive systematic convolutional (RSC) code with generator matrix $G = [1,\ \frac{1+D^2}{1+D+D^2}]$. It is convenient to choose the SF to be $H^T = [\frac{1+D^2}{1+D+D^2},\ 1]^T$ and the ISF to be $(H^{-1})^T = [0,\ 1]$. Clearly, constraints (7) and (8) are satisfied since $[1,\ \frac{1+D^2}{1+D+D^2}][\frac{1+D^2}{1+D+D^2},\ 1]^T = \frac{1+D^2}{1+D+D^2} \oplus \frac{1+D^2}{1+D+D^2} = 0$ and $[0,\ 1][\frac{1+D^2}{1+D+D^2},\ 1]^T = 1$. The encoder, the SF and the ISF of this convolutional code are illustrated in Fig 3. Another valid SF-ISF pair is to choose $H^T = [1+D^2,\ 1+D+D^2]^T$ and $(H^{-1})^T = [1+D,\ D]$.

B. Parallel Concatenated Convolutional Codes

Consider a parallel turbo code with two RSC component codes connected by a random interleaver. Assume that the first component code has rate $R_1 = k/n_1$ and generator matrix $G_1 = [I,\ P_1]$, where I is the $k \times k$ identity matrix to generate k systematic bits and P_1 is a $k \times (n_1 - k)$ matrix to generate $(n_1 - k)$ parity check bits for every block of k input bits. Similarly, assume that the second component code has rate $R_2 = k/n_2$ and generator matrix $G_2 = [I,\ P_2]$ where P_2 is a $k \times (n_2 - k)$ matrix to generate $(n_2 - k)$ check bits for every k input bits. For each block, the systematic bits and parity check bits from the G_1 and G_2 are denoted by $\mathbf{y}_s$,$\mathbf{y}_1$, $\tilde{\mathbf{y}}_s$,$\mathbf{y}_2$ respectively. Since $\tilde{\mathbf{y}}_s$ is a scrambled version of $\mathbf{y}_s$, it is not transmitted. Hence, the overall rate for the concatenated code is $R = k/(n_1 + n_2 - k) = R_1 R_2/(R_1 + R_2 - R_1 R_2)$.

The syndrome sequence of the overall parallel turbo code can be formed of two parts $\mathbf{s} = [\mathbf{s}_1,\ \mathbf{s}_2]$ where $\mathbf{s}_1$ comes from the first component code and $\mathbf{s}_2$ the second. Due to the random interleaver, the overall syndrome former of the turbo code does not have a simple closed-form expression. Nevertheless, by exploiting the original structure of the turbo encoder, a simple SF in the form of linear sequential circuit can still be implemented with turbo codes.

Following the discussion in the previous subsection, we can easily obtain the (sub) syndrome formers for the two RSC component codes. An obvious choice is to choose

$$H_1^T = [P_1,\ I]^T, \tag{9}$$

$$\text{and} \qquad H_2^T = [P_2,\ I]^T. \tag{10}$$

The overall SF is then formed by parallel concatenating the two sub SFs with a random interleaver in a form similar to the original turbo encoder. This structure is illustrated in Fig. 4. Clearly, for every $(n_1 + n_2 - k)$ bits at the input of the SF, H_1^T outputs $(n_1 - k)$ bits and H_2^T outputs $(n_2 - k)$ bits. The overall SF thus has a transfer matrix operating on $(n_1 + n_2 - k)$ and

[1]It should be noted that the generator matrix of a binary convolutional code considered in this work is formed of generator polynomials and, hence, is different from the $\{0, 1\}$ generator matrices of a block code.

producing $(n_1 + n_2 - 2k)$ bits as required. It is also easy to verify that for all valid codewords, the SF outputs the all-zero syndrome $\mathbf{0}$. Notice that if the interleaver is fixed, this SF is still a linear circuit. Hence, it is a simple and valid syndrome former for the subject parallel turbo code.

The role of the ISF is, for a given syndrome $\mathbf{s}$, to find an auxiliary sequence (not necessarily a valid codeword of the channel code) that is associated with this syndrome. For each of the RSC component codes, the (sub) ISF that performs a left inverse of the (sub) SF will output a sequence that contains both a systematic part and a parity part, i.e, $[\mathbf{y}_s,\ \mathbf{y}_1]$ from sub ISF1 and $[\tilde{\mathbf{y}}_s,\ \mathbf{y}_2]$ from sub ISF2. Due to the random interleaver, the bit sequence fed into the second component code is a scrambled version of what is fed into the first one, causing a potential misalignment of the systematic bits $\mathbf{y}_s$ and $\tilde{\mathbf{y}}_s$. This seems to suggest the need for some form of interleaving/deinterleaving at the decoder which can be tricky and complex[2]. However, a bit more thought reveals a simple and effective solution, that is, to force the systematic part of the ISF to always output all-zero sequences (which are invariant regardless of what interleaver is used). Hence, the choice of the sub ISF for the component RSC code is restricted to $(H_{1,2}^{-1})^T = [\mathbf{0},\ J]$, where the left part is a zero matrix and right part J is a square matrix. Specifically, for the sub SFs we select (9) and (10) and the corresponding sub ISFs take the form of $[\mathbf{0},\ I]$. Implied in this practice is that for any syndrome/coset-leader, there exists one and only one codeword/sequence which falls in this bin/coset and which has all-zero systematic part. This can be easily shown to be valid for linear codes. The general structure of the ISF of the overall turbo code is shown in Fig. 4.

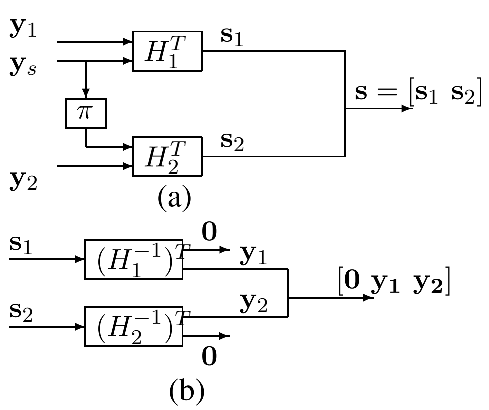

Fig. 4. (a) General SF for PCCC. (b) General ISF for PCCC

Now that we have constructed the SF and the ISF for parallel turbo codes, the rest of the work with source coding can follow the steps discussed in Section II. For the readers' convenience, we summarize the whole process as follows:

- **Parallel Turbo Code:** the turbo code in use has two RSC component codes with generator matrices of $G_1 = [I,\ P_1]$ and $G_2 = [I,\ P_2]$, respectively.

[2]For example, in the implicit turbo-binning scheme in [16], auxiliary bits (and parallel branches) are introduced to the source coding trellis to handle the random scrambling in a turbo code.

- **Syndrome Former:** the SF is formed of an interleaved parallel concatenation of two sub SFs corresponding to the two component codes (Fig. 4).
- **Inverse Syndrome Former:** the ISF is formed by parallel concatenating the two corresponding sub ISFs, $(H_1^{-1})^T = [\mathbf{0},\ J_1]$ and $(H_2^{-1})^T = [\mathbf{0},\ J_2]$ (Fig. 4).
- **Encoder:** the encoder is nothing but a syndrome former. The source sequence at the input is viewed as a three-segment virtual codeword of the turbo code: $\mathbf{x} = [\mathbf{x}_s,\ \mathbf{x}_1,\ \mathbf{x}_2]$, where $\mathbf{x}_s$, $\mathbf{x}_1$, $\mathbf{x}_2$ are treated as virtual systematic bits and virtual parity bits from the first and second component code, respectively. The source sequence $\mathbf{x}$ is passed into the SF, compressed to a syndrome sequence, $\mathbf{s} = [\mathbf{s}_1,\ \mathbf{s}_2]$ which is sent over the channel.
- **Channel:** send $\mathbf{s} = [\mathbf{s}_1,\ \mathbf{s}_2]$ to decoder without error i.e., noiseless channel.
- **Decoder**: the decoder (Fig. 5) is composed of an ISF and a conventional turbo decoder. The auxiliary sequence at the output of the ISF is subtracted from the side information, $\mathbf{y}$, and then fed into the turbo decoder. At the output of the turbo decoder, the auxiliary sequence is added back to recover the original sequence $\mathbf{x}$ as $\hat{\mathbf{x}}$.

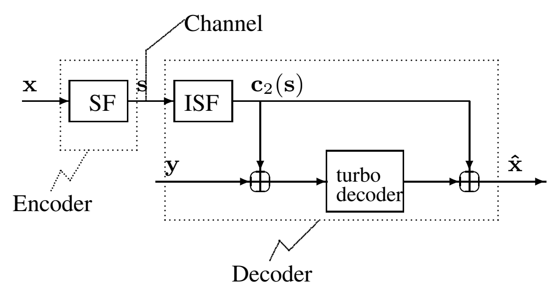

Fig. 5. System diagram for source coding with side information

C. Serial Concatenated Convolutional Codes

Following the same line of thinking as in PCCC codes, SCCC codes can also be exploited for compression with side information. Due to the space limitation, we only pinpoint the key steps here. Interested readers please refer to [20]. Again, the SF-ISF pair of a serial turbo code is based on the SF-ISF pairs of its component codes. To avoid the potential misalignment caused by the random interleaver, it is convenient to restrict the inner code to be systematic. The syndrome of the overall serial code can be decomposed of two parts $\mathbf{s} = [\mathbf{s_1},\ \mathbf{s_2}]$ where $\mathbf{s_1}$ comes from the inner code and $\mathbf{s_2}$ the outer code. $\mathbf{s_1}$ can be obtained by applying the codeword to the SF of the inner code, and $\mathbf{s_2}$ by passing the de-scrambled systematic part of codeword to the SF of the outer code. The auxiliary sequence, i.e. the output of the ISF, can be obtained from the binary addition of two parts: (i) the output from the sub ISF of the inner code (with $\mathbf{s_1}$ as the input); It should be noted that this sub ISF needs to force the systematic part of the output to be all zeros. (ii) the output of the inner encoder fed with the scrambled output of the sub ISF of the outer code (with $\mathbf{s_2}$ as the input).

IV. Simulation Results

The proposed scheme represents a direct exploitation of the binning/coset/syndrome approach discussed in Section II, and is immediately applicable to any parallel turbo code and many serial turbo codes. To evaluate its performance, we simulate the proposed binning scheme on a rate-1/3, 8-state parallel turbo code with the same component codes as in [12][16]: $G_1 = G_2 = [1, \frac{1+D+D^2+D^3}{1+D^2+D^3}]$. A length 10^4 S-random interleaver with a spreading factor 17 and a length 10^3 S-random interleaver with a spreading factor 11 are used for simulation, and ten decoding iterations are performed before the turbo decoder outputs its estimates. Appropriate clip-values are also used to avoid numerical overflows/downflows in the turbo decoder. Table I lists the simulation results where n denotes the interleaver length and for each crossover probability p, 3×10^7 bits are simulated except for $p = 0.14/0.145$, $n = 10^4$ where 3×10^9 bits are simulated. The interleaving gain can be easily seen from the table. If a normalized distortion of 10^{-6} is considered near-lossless, then this turbo coding scheme can support a virtual BSC with $p = 0.145$. Since the compression rate is 2/3, there is a gap of only $2/3 - H(0.145) = 0.07$ from the theoretic limit, which is among the closest gaps reported so far. We mention that in [12] and [16], the same turbo code with the same interleaver size but different code rate is used. The achievable performances therein are about 0.15 and 0.09 from the theoretic limit, respectively.

In addition to conventional binary turbo codes, asymmetric turbo codes which employ a different component code at each branch can also be applied for capacity-approaching DSC. Asymmetric turbo codes bear certain advantages in joint optimizing the performance at both the water-fall region and the error floor region [17].

We simulated the NP16-P16 code in [17] with $G_1 = [1, \frac{1+D^4}{1+D+D^2+D^3+D^4}]$ and $G_2 = [1, \frac{1+D+D^2+D^4}{1+D^3+D^4}]$. A length 10^4 S-random interleaver with a spreading factor 17 is applied and 15 turbo decoding iterations are performed. Simulation results show that the proposed turbo-binning scheme provides a distortion of 5.6×10^{-7} when $p = 0.15$. This translates to a gap less than 0.06 from the theoretic limit.

V. Conclusion

We have proposed an efficient turbo-binning scheme for the noiseless DSC problem using a syndrome former and an inverse syndrome former. The proposed approach is simple, optimal and widely applicable. Computer simulations reveal good performance which is close to the theoretic limit. With this proposed turbo-binning scheme in mind, implementing turbo codes for the noiseless DSC problem is simplified to two steps: (i) Choose an appropriate parallel/serial turbo code for the virtual BSC which models the correlation between the two binary sources, and compress one source with a conventional compression method. (ii) Follow the scheme we outlined to construct an efficient SF-ISF pair for this code, and subsequently to encode (i.e. compress) and decode (i.e. decompress) the other source using the binning approach.

Aside from simplicity, a particularly nice feature about the proposed turbo-binning scheme is its unaltered use of an existing parallel turbo code (no redesign of the code structure or decoding algorithm is needed). This allows the rich results available in the literature of turbo codes to serve immediately and directly the DSC problem at hand.

TABLE I

Performance of the Proposed Turbo-Binning Scheme

Crossover Prob. p	Distortion $n = 10^3$	Distortion $n = 10^4$
0.10	0	0
0.11	1.5×10^{-6}	0
0.14	8.0×10^{-4}	4.1×10^{-7}
0.145	4.0×10^{-3}	6.4×10^{-7}
0.155	3.5×10^{-2}	4.2×10^{-3}

References

[1] A. D. Wyner. Recent results in the shannon theory. *IEEE Trans. Inform. Theory*, pages 2–10, Jan. 1974.

[2] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory*, pages 471–480, July 1973.

[3] Y. Oohama and T. S. Han. Universal coding for the slepian-wolf data compression system and the strong converse. *IEEE Trans. Inform. Theory*, 40:1908–1919, Nov. 1994.

[4] A. Wyner. On source coding with side information at the decoder. *IEEE Trans. Inform. Theory*, 21:294–300, May 1975.

[5] S. Shamai and S. Verdu. Capacity of channels with side information. *European Trans. Telecommun*, 6:587–600, Sep.-Oct. 1995.

[6] R. Zamir, S. Shamai, and U. Erez. Nested linear/lattice codes for structured multiterminal binning. *IEEE Trans. Inform. Theory*, pages 1250–1276, June 2002.

[7] S. S. Pradhan and K. Ramchandram. Distributed source coding using syndromes (DISCUS): Design and construction. *IEEE Tran. Inform. Theory*, pages 626–643, Mar. 2003.

[8] S. Servetto. Quantization with side information: lattice codes, asymptotics, and applications in wireless networks. *IEEE Trans. Inform. Theory, submitted to*, 2002.

[9] A. Liveris, Z. Xiong, and C. N. Georghiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Communications Letters*, 6:440–442, Oct. 2002.

[10] A. Liveris, Z. Xiong, and C. N. Georghiades. Compression of binary sources with side information using low-density parity-check codes. *Proc. GLOBECOM*, pages 1300–1304, Nov. 2002.

[11] R. Hu, R. Viswanathan, and J. Li. A new coding scheme for the noisly-channel slepian-wolf problem: Separate design and joint decoding. *Proc. IEEE GLOBECOM*, 2004.

[12] J. Garcia-Frias and Y. Zhao. Compression of correlated binary sources using turbo codes. *IEEE Commun. Letters*, pages 417–419, Oct. 2001.

[13] A. Aaron and B. Girod. Compression with side information using turbo codes. *Proc. of IEEE Data Compression Conference (DCC)*, April 2002.

[14] J. Bajcsy and P. Mitran. Coding for the slepian-wolf problem with turbo codes. *Proc. IEEE GLOBECOM*, Nov 2001.

[15] J. Li, Z. Tu, and R. S. Blum. Slepian-wolf coding for nonuniform sources using turbo codes. *Proc. IEEE Data Compression Conf.*, pages 312–321, March 2003.

[16] A. D. Liveris, Z. Xiong, and C. N. Georghiades. Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes. *Proc. of IEEE Data Compression Conference (DCC)*, Mar. 2003.

[17] O. Y. Takeshita, O. M. Collins, P. C. Massey, and D. J. Costello Jr. A note on asymmetric turbo codes. *IEEE Comm. Letters*, 3:69–71, March 1999.

[18] I. Csiszar. Linear codes for sources and source networks: error exponents, universal coding. *IEEE Trans. Inform. Theory*, 28:585–592, July 1982.

[19] Jr G. D. Forney. Trellis shaping. *IEEE Trans. Inform. Theory*, pages 281–300, Mar. 1992.

[20] Z. Tu, J. Li, and R. S. Blum. An efficient sf-isf approach for the Slepian-Wolf source coding problem. *submitted to Eurasip J. on Applied Signal Processing - Speical Issue on Turbo Processing*, 2003.