

# ROBUST AND FAST NON ASYMMETRIC DISTRIBUTED SOURCE CODING USING TURBO CODES ON THE SYNDROME TRELLIS

V. Toto-Zarasoá, A. Roumy, C. Guillemot and C. Herzet

INRIA, Campus Universitaire de Beaulieu, 35042 Rennes-Cedex, France.

## ABSTRACT

We consider the distributed compression of two (binary memoryless) correlated sources and propose a unique codec that can reach any point in the Slepian-Wolf region. In a previous method based on channel codes, the decoder multiplies the compressed data by an inverse submatrix of the code. This multiplication presents two drawbacks. First, if turbo codes are used, the submatrix has no periodic structure s.t. the whole inverse has to be stored and no fast implementation exists for the multiplication. Second, this multiplication may lead to error propagation. In this paper, we propose a method that is both robust and fast.

**Index terms**— Distributed source coding, channel coding, linear block codes, turbo codes.

## 1. INTRODUCTION

Distributed source coding (DSC) refers to the problem of disjoint compression of correlated sources. It was funded in 1973 by the Slepian-Wolf (SW) theorem [1]. In the following, lossless compression of two correlated binary sources is considered.

It has long been known, thanks to Shannon's theorem, that cooperation between the sources leads to a minimum achievable rate amounting to the joint entropy. If cooperation is not allowed between the two sources, the Slepian-Wolf theorem states the surprising result that the joint entropy is still the minimum achievable rate, as long as each single rate is greater than the conditional entropy, and joint decoding is performed. The set of achievable rates is called the SW region.

This result is particularly valuable in the domain of wireless sensor networks [2] where neighboring and thus highly correlated sensors do not need to communicate to reach optimal compression rates. Moreover, to adapt to channel condition variations between the sources and the receiver [3], the encoders should dynamically vary their compression rates in order to reach any point of the SW region. It is the construction of such a rate adaptive code that we consider in this paper.

In [4], Wyner showed the optimality of binary linear codes to reach a corner point of the SW region when two binary

This research was partially supported by the French National Research Agency (ANR) under the Essor project and by the French European Commission in the framework of the FP7 Network of Excellence in Wireless Communications NEWCOM++ (contract n.216715).

memoryless sources have to be compressed. This scheme is usually referred to as the *syndrome approach*. Therefore, first practical solutions were based on channel codes [5, 6]. Here we consider the syndrome approach proposed in [7] that can deal with any convolutional code (not only systematic as in [5]) and any turbo code. We propose a construction that allows to reach any point of the SW region based on the scheme [8, 9], where LDPC codes were used. If turbo codes are considered, there is no fast implementation for the scheme [8, 9] and propagation errors occur. In this contribution, we propose a *robust* and also *fast* non asymmetric SW scheme.

## 2. THE DSC PROBLEM

Let  $X$  and  $Y$  be two correlated sources having realizations  $\mathbf{x}$  and  $\mathbf{y}$ .  $R_x$  and  $R_y$  are their respective compression rates. To recover both sources in the SW setup, the compression rates must satisfy:  $R_x \geq H(X|Y)$ ,  $R_y \geq H(Y|X)$  and  $R_x + R_y \geq H(X, Y)$ . These inequalities define the so called SW region depicted in Fig. 1.

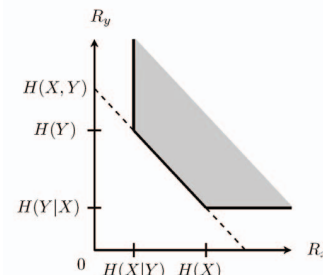


Fig. 1. The Slepian-Wolf rate region.

**Asymmetric DSC** corresponds to a SW codec that operates at a corner point of the SW region. In [4], Wyner showed that binary linear codes can achieve these corner points with a scheme called the *syndrome approach*. More precisely, consider an  $(n, k)$  channel code  $\mathcal{C}$  defined by its  $(n - k) \times n$  parity check matrix  $\mathbf{H}$ :  $\mathcal{C} = \{\mathbf{x} : \mathbf{x}\mathbf{H}^T = \mathbf{0}\}$ . This code defines a partition of the  $n$ -length sequences into cosets, where all sequences in a coset share the same syndrome:  $\mathcal{C}_s = \{\mathbf{x} : \mathbf{x}\mathbf{H}^T = \mathbf{s}\}$ . To encode  $\mathbf{x}$ , the encoder transmits its syndrome  $\mathbf{s}_x = \mathbf{x}\mathbf{H}^T$  achieving a compression ratio of  $n : (n - k)$ . At the decoder,  $\mathbf{y}$  have been sent at its entropy rate  $H(Y)$  and can therefore be retrieved. Then, the sequence  $\mathbf{x}$  is found as the closest sequence to  $\mathbf{y}$  with syndrome  $\mathbf{s}_x$ . Since the

search is performed in any coset and not only the set of code-words (with syndrome  $\mathbf{0}$ ), classical channel decoder have to be adapted. This adaptation has been proposed in [5, 10] for systematic convolutional codes and systematic turbo codes and for LDPC codes in [6]. We consider the decoder in [7] that can deal with any convolutional and turbo code (not only systematic).

**Non asymmetric DSC** is the problem we address in this paper. It aims at achieving any point between the two corner points of the SW region (see Fig. 1). Two non asymmetric schemes have been proposed: in [2, 11], the asymmetric code is partitioned into two subcodes, one for each source, whereas the original code is used in [8, 9]. In both approaches, first the difference pattern between the two sources  $\mathbf{z} = \mathbf{x} \oplus \mathbf{y}$  is estimated, then the sources are recovered. In this two step procedure, error propagation can occur if the error pattern  $\mathbf{z}$  is not correctly estimated.

### 3. NON ASYMMETRIC SW CODING

#### 3.1. Principle

In this section, we review the general principle of non asymmetric SW coding proposed in [8, 9] and applied to LDPC codes. This scheme uses the  $(n - k) \times n$  parity-check matrix  $\mathbf{H}$  defined above and a design parameter  $k' \in [0, k]$ , that allows to vary the compression rates  $(R_x, R_y)$ . The two correlated binary sequences of length  $n$ ,  $\mathbf{x}$  and  $\mathbf{y}$ , are split into three parts. More precisely,  $\mathbf{x} = (\mathbf{x}_1^{k'}, \mathbf{x}_{k'+1}^k, \mathbf{x}_{k'+1}^n)$  and  $\mathbf{y} = (\mathbf{y}_1^{k'}, \mathbf{y}_{k'+1}^k, \mathbf{y}_{k'+1}^n)$ . The syndromes  $\mathbf{s}_x = \mathbf{x}\mathbf{H}^T$  and  $\mathbf{s}_y = \mathbf{y}\mathbf{H}^T$ , of length  $(n - k)$ , are computed for both sequences and transmitted to the decoder. In addition, the  $k'$  first bits of  $X$ ,  $\mathbf{x}_1^{k'}$ , and the  $k - k'$  next bits for the source  $Y$ ,  $\mathbf{y}_{k'+1}^k$ , are transmitted as systematic bits. The total rate for the sequences  $\mathbf{x}$  and  $\mathbf{y}$  of length  $n$  is respectively  $n - k + k'$  and  $n - k'$  bits. The structure of the coders is depicted in Figure 2.

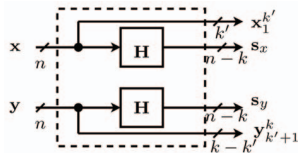


Fig. 2. The non asymmetric coders in the syndrome approach.

At the receiver, the decoder first estimates the error pattern  $\mathbf{z} = \mathbf{x} \oplus \mathbf{y}$ . By definition, the MAP estimate of the error pattern (based on the knowledge of  $\mathbf{s}_x$  and  $\mathbf{s}_y$ ) is the minimum weight vector with syndrome  $\mathbf{s}_z = \mathbf{s}_x \oplus \mathbf{s}_y$ . In other words, it is the closest sequence to the zero sequence  $\mathbf{0}$ , with syndrome  $\mathbf{s}_z$ . It can thus be estimated with any asymmetric SW decoder as shown in Figure 3.

Once the error pattern is found, the subsequences of information bits  $\mathbf{x}_{k'+1}^k$  and  $\mathbf{y}_1^{k'}$  can be retrieved from the error pattern  $\mathbf{z} = \mathbf{x} \oplus \mathbf{y}$  as:

$$\hat{\mathbf{x}}_{k'+1}^k = \mathbf{y}_{k'+1}^k \oplus \hat{\mathbf{z}}_{k'+1}^k \text{ and } \hat{\mathbf{y}}_1^{k'} = \mathbf{x}_1^{k'} \oplus \hat{\mathbf{z}}_1^{k'} \quad (1)$$

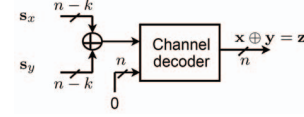


Fig. 3. Non-asymmetric decoder: estimation of the error pattern  $\mathbf{z} = \mathbf{x} \oplus \mathbf{y}$ .

Subsequences of  $n - k$  bits, i.e.  $\mathbf{x}_{k'+1}^n$  and  $\mathbf{y}_{k'+1}^n$ , remain to be computed for both sequences. Let us assume that  $\mathbf{H} = (\mathbf{A} \ \mathbf{B})$ , where  $\mathbf{B}$  is an invertible square matrix of dimension  $(n - k) \times (n - k)$ . Such a decomposition (up to a permutation on the columns) exists since  $\mathbf{H}$  has rank  $n - k$ . Thus,  $\mathbf{s}_x = \mathbf{x}\mathbf{H}^T = \mathbf{x}(\mathbf{A} \ \mathbf{B})^T = \mathbf{x}_1^k \mathbf{A}^T \oplus \mathbf{x}_{k'+1}^n \mathbf{B}^T$ , and the remaining  $n - k$  unknown bits of the sequence  $\mathbf{x}$  (and similarly for the sequence  $\mathbf{y}$ ) can be computed as

$$\hat{\mathbf{x}}_{k'+1}^n = (\mathbf{s}_x \oplus \hat{\mathbf{x}}_1^k \mathbf{A}^T) \mathbf{B}^{-1 T}, \quad (2)$$

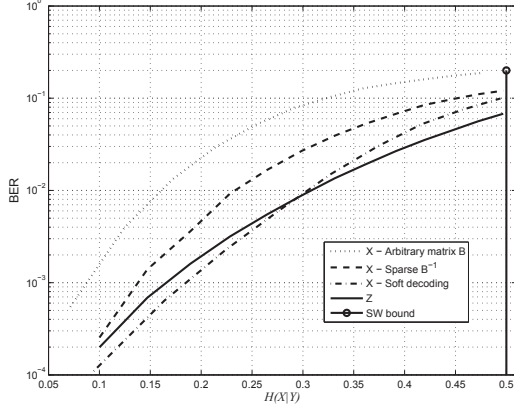
where  $\mathbf{B}^{-1}$  denotes the inverse of the matrix  $\mathbf{B}$ .

This scheme has two drawbacks. First, if turbo or LDPC codes are used, the inverse submatrix of the code,  $\mathbf{B}^{-1}$ , has no periodic structure s.t. the whole matrix  $\mathbf{B}^{-1}$  has to be stored, and the complexity of the multiplication grows quadratically with the blocklength. Second, from the construction of the decoder, we notice that the three parts of the estimated sequences  $\hat{\mathbf{x}}$  and  $\hat{\mathbf{y}}$  have different characteristics. Since the subsequences  $\mathbf{x}_1^{k'}$  and  $\mathbf{y}_{k'+1}^k$  are available at the decoder, their estimates have no error. The next subsequences ( $\mathbf{x}_{k'+1}^k$  and  $\mathbf{y}_1^{k'}$ ) are recovered thanks to the estimated error pattern (1), therefore their estimates have the same BER as the estimated error pattern. The most critical issue that we address in this paper is the BER of the  $n - k$  unknown bits of the sequence  $\mathbf{x}$  (and similarly for  $\mathbf{y}$ ). The multiplication by the matrix  $\mathbf{B}^{-1}$  in (2) may enhance the number of errors. We call this effect the *error propagation phenomena*. In the following, we show this effect for the convolutional and turbo codes and propose solutions to limit it. Moreover, we will show that this robust scheme is also fast in the sense that the complexity grows only linearly with the blocklength.

#### 3.2. Non asymmetric SW coding using convolutional codes

In the following, we consider a  $(n, k)$  convolutional code with  $N$  periods. First, we note that the unknown positions in the vector  $\mathbf{x}$  are design parameters. More precisely, these positions (or equivalently the columns of the matrix  $\mathbf{H}$  to be extracted in order to build the matrix  $\mathbf{B}$ ) are chosen at the encoder and known at the decoder. Therefore, the estimation of the subsequences  $\mathbf{x}_{k'+1}^n$  and  $\mathbf{y}_{k'+1}^n$  is an easier task than channel decoding over a binary erasure channel (BEC), since, in our case, the erased positions are known at the encoder.

Figure 4 shows the BER of the error pattern  $\mathbf{z}$  (continuous line) and its effect on the estimation of the source sequence  $\mathbf{x}$ . Performance is shown for a convolutional code. We consider three possible estimators.

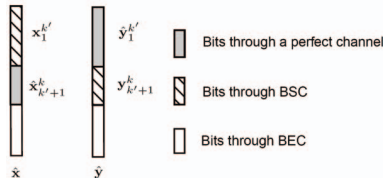


**Fig. 4.** Error propagation in the estimation of the source  $X$ . The convolutional code is defined by its parity check matrix  $\mathbf{H} = \begin{pmatrix} 11 & 15 & 06 \\ 15 & 12 & 17 \end{pmatrix}$  and is punctured over a period of four trellis sections in order to get a 2 : 1 compression rate for the source.

First, we use the original method (2). The dotted curve represents the BER, when equation (2) is performed with an arbitrary invertible matrix  $\mathbf{B}$ . As expected, the BER has been drastically increased. Since the linear code chosen is a convolutional code, its parity check matrix has a natural periodic structure. Therefore, there exists a sliding window implementation for computing (2), which reduces the required storage (a whole matrix  $\mathbf{B}^{-1}$  does not need to be stored).

In a second method, we optimize the matrix  $\mathbf{B}$  in order to limit the error propagation. More precisely, we choose a matrix  $\mathbf{B}$  which inverse is as sparse as possible. Fig. 4 shows that if the matrix is chosen to be as sparse as possible, then the BER can be lowered from the dotted curve to the dashed curve. Here, an exhaustive search over all possible  $\mathbf{B}$  matrices has been performed.

Finally, we use a modified ML decoder to solve equation (2). Let us first assume that the error pattern  $\mathbf{z}$  is perfectly known at the decoder. Therefore the  $k$  first bits of  $\mathbf{x}$ ,  $\mathbf{x}_1^k$ , are also known. If the matrix  $\mathbf{B}$  is invertible, the original source sequence is the unique sequence of the coset  $\mathcal{C}_{\mathbf{s}_x}$  that has the first  $k$  bits equal to  $\mathbf{x}_1^k$  (recall that the syndrome  $\mathbf{s}_x$  is also known at the decoder). Therefore, one can build a ML decoder (Viterbi for the convolutional decoder) that performs a search in the coset  $\mathcal{C}_{\mathbf{s}_x}$  and that is matched to a channel combining a perfect channel (for  $\mathbf{x}_1^k$ ) and an erasure channel (for  $\mathbf{x}_{k+1}^n$ ). The effect of knowing some bits perfectly is that some paths are deleted in the trellis, and if  $\mathbf{B}$  is invertible, there is even a single path that remains.



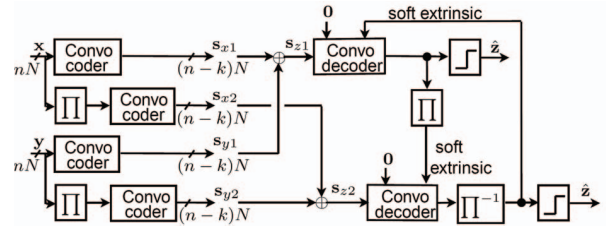
**Fig. 5.** Equivalent channels for the decoding of the sources.

We now go back to our original problem of error propagation. If the difference pattern  $\mathbf{z}$  contains errors, the decoder is

matched to a channel combining a perfect channel (for  $\mathbf{x}_1^k$ ), a BSC (for  $\hat{\mathbf{x}}_{k'+1}^k$  with cross over probability the BER of  $\mathbf{z}$ ) and an erasure channel (for  $\mathbf{x}_{k+1}^n$ ) as shown in Fig. 5. Fig. 4 shows that the BER for the estimation of  $\mathbf{x}$  is further reduced (dot-dash curve). Interestingly, with this decoder the BER of  $\mathbf{x}$  remains almost the same as that of  $\mathbf{z}$ . The proposed can therefore limit the error propagation. Moreover, the complexity of the proposed algorithm grows linearly with the block-length  $Nn$ .

### 3.3. Non asymmetric SW coding using turbo codes

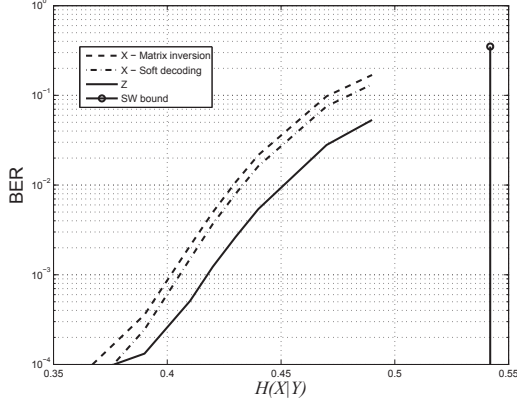
In this section, we use a turbo code composed of two identical  $(n, k)$  convolutional codes (of period  $N$ ) separated by a random interleaver of size  $Nn$ , as shown in Fig. 6. Each source transmits two syndromes of length  $(n - k)N$  each (one of the source:  $\mathbf{s}_{x1}$  ( $\mathbf{s}_{y1}$ ), and one of the interleaved version of the source:  $\mathbf{s}_{x2}$  ( $\mathbf{s}_{y2}$ )).



**Fig. 6.** The turbo-syndrome scheme for non asymmetric coding of two sources: the encoder and the estimation of the error pattern.

As presented in Section 3.1, the first step consists in estimating the error pattern  $\mathbf{z}$ . Therefore, we compute the two syndromes of  $\mathbf{z}$ :  $\mathbf{s}_{zi} = \mathbf{s}_{xi} \oplus \mathbf{s}_{yi}$ ,  $i \in \{1, 2\}$  and search for the closest sequence to  $\mathbf{0}$  in  $\mathcal{C}_{\mathbf{s}_{z1}} \cap \mathcal{C}_{\mathbf{s}_{z2}}$ . To perform this search, we use the modified BCJR algorithm [7] for each convolutional decoding, that passes soft extrinsic messages between the convolutional decoders as depicted in Fig. 6. The turbo decoding stops when the estimated  $\hat{\mathbf{z}}$  matches the two syndromes  $\mathbf{s}_{z1}$ ,  $\mathbf{s}_{z2}$  or when a maximum number of turbo decoding loops is reached.

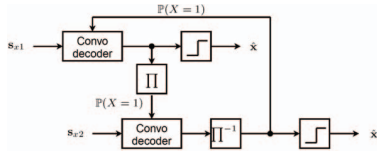
Once  $\hat{\mathbf{z}}$  is estimated, the source sequences  $\mathbf{x}$  and  $\mathbf{y}$  have to be recovered as in (2). The first method considered here consist in computing the parity check matrix of the turbo code. Let  $\mathbf{H}_1$  be the  $N(n - k) \times Nn$  parity check matrix defining the first convolutional code. The parity check matrix of the second code  $\mathbf{H}_2$  results from a permutation of the columns of  $\mathbf{H}_1$ . Thus, the turbo code is completely determined by the  $2N(n - k) \times Nn$  matrix  $\mathbf{H}^T = (\mathbf{H}_1^T \mathbf{H}_2^T)$ . Given a decomposition  $\mathbf{H} = (\mathbf{A} \mathbf{B})$  with an invertible  $\mathbf{B}$ , we apply (2) to estimate the sources  $X$  and  $Y$ . Due to the presence of the interleaver, there exists no sliding window implementation of this estimator as for convolutional codes (unless a constrained interleaver is designed which may degrade the performance of the turbo code). Moreover, as expected, the error propagates drastically. Therefore, we propose a novel method that can both reduce the error propagations and have a sliding window implementation (to have an estimator complexity that grows linearly with  $N$ ).



**Fig. 7.** Error propagation and estimation of the source  $X$ . Each constituent code of the turbo code is defined by  $\mathbf{H} = \begin{pmatrix} 11 & 15 & 06 \\ 15 & 12 & 17 \end{pmatrix}$  and is punctured over a period of 4 trellis sections in order to get a 2:1 compression rate. The blocksize is 2004.

The estimation of the remaining subsequences proposed for convolutional codes rely on the hypothesis that an exact MAP decoding can be used. However, for turbo codes such a decoder is of great complexity and we consider here the usual suboptimal MAP decoder (see Fig. 8). As for the case of convolutional codes, we first assume that the  $\mathbf{z}$  pattern has been perfectly estimated. Each constituent decoder has to solve a system of  $N(n - k)$  equations with  $2N(n - k)$  unknowns and the solution is “at best” a vector subspace of dimension  $N(n - k)$ . To insure that the dimension of each solution subspace is only  $N(n - k)$ , the matrices  $\mathbf{B}_{11}$  and  $\mathbf{B}_{22}$  must be invertible, where these matrices result from the decomposition of  $\mathbf{B}$  into  $\mathbf{B} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{pmatrix}$ . Finally, to insure that the intersection of the two solution subspaces reduces to a single sequence the whole matrix  $\mathbf{B}$  has also to be invertible.

These three new constraints (existence of  $\mathbf{B}^{-1}$ ,  $\mathbf{B}_{11}^{-1}$  and  $\mathbf{B}_{22}^{-1}$ ) are enough if the two decoders communicate their solution subspaces to each other. However, for a low complexity decoder implementation, we rely on a *bit* MAP decoding procedure that may lose some information on the solution subspace. More precisely, no information is lost if all the erased bits of a solution subspace are located in the same position or in other words if a solution subspace is generated by coordinate vectors (the  $i$ -th coordinate vector has a single 1 at the  $i$ -th position and 0 everywhere else  $\mathbf{e}_i = (0 \dots 010 \dots 0)$ ). If such a condition is not satisfied, then the turbo decoder has to be helped by introducing some known bits at the decoder.



**Fig. 8.** The turbo-syndrome scheme for the non asymmetric decoding of one source.

Figure 8 shows the BER of the error pattern  $\mathbf{z}$  (continuous line) and its effect on the estimation of the source sequence

x. The turbo code considered has a global compression rate of 2:1. However, some unknown bits of the source are directly sent to the decoder (for both estimators of the source  $X$ ). Therefore, the global compression rate (and thus the SW bound in the figure) increases to 0.54. First, the matrix inversion technique (2) is used and the dashed curve shows that the BER has been increased. Finally, the source  $X$  is estimated with the soft decoder described above and the dot-dash curve shows that the BER has been reduced. Interestingly, our robust scheme is also fast since the complexity grows linearly with the interleaversize  $Nn$ .

#### 4. CONCLUSION

We have proposed a non asymmetric SW codec, for two correlated binary memoryless sources, constructed with convolutional or turbo codes. First we have shown the existence of propagation error between the estimated difference pattern (difference of the two sources) and the reconstruction of the sources. Then, we have proposed a robust scheme which limits the error propagation effect. Finally, the proposed scheme has a natural sliding window implementation s.t. the complexity grows linearly with the blocksize.

#### 5. REFERENCES

- [1] D. Slepian and J. K. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, July 1973.
- [2] S. S. Pradhan and K. Ramchandran, “Distributed source coding: symmetric rates and applications to sensor networks,” in *Proceedings of the IEEE Data Compression Conference*, March 2000, pp. 363–372.
- [3] A. Roumy and D. Gesbert, “Optimal matching in wireless sensor networks,” *IEEE Journal on Selected Topics in Signal Processing, Special issue on Convex Optimization Methods in Signal Processing*, December 2007.
- [4] A. Wyner, “Recent results in the Shannon theory,” *IEEE Transactions on Information Theory*, vol. 20, pp. 2–10, 1974.
- [5] S. S. Pradhan and K. Ramchandran, “Distributed source coding using syndromes (discus): Design and construction,” *Proceedings of the IEEE Data Compression Conference*, pp. 158–167, March 1999.
- [6] A. D. Liveris, Z. Xiong, and C. N. Georghiades, “Compression of binary sources with side information at the decoder using ldpc codes,” *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, October 2002.
- [7] A. Roumy, K. Lajnef, and C. Guillemot, “Rate-adaptive turbo-syndrome scheme for slepian-wolf coding,” in *41st Asilomar Conference on Signals, Systems and Computers*, November 2007.
- [8] N. Gehrig and P. L. Dragotti, “Symmetric and asymmetric slepian-wolf codes with systematic and non-systematic linear codes,” *IEEE communications letters*, vol. 9, no. 1, pp. 61–63, January 2005.
- [9] P. Tan and J. Li, “A practical and optimal symmetric slepian-wolf compression strategy using syndrome formers and inverse syndrome formers,” in *Proceedings of 43rd Annual Allerton Conference on Communication, Control and Computing*, September 2005.
- [10] Z. Tu, J. Li, and R. S. Blum, “An efficient SF-ISF approach for the Slepian-Wolf source coding problem,” *Eurasip Journal on Applied Signal Processing*, 6(0):961–971, May 2005.
- [11] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiades, “Design of slepian-wolf codes by channel code partitioning,” in *Proceedings of the IEEE Data Compression Conference*, 2004, pp. 302–311.