IEEE Canadian Workshop on Information Theory, Edmonton, Alberta, Canada, 2007.

A Variable Rate Slepian-Wolf Code Construction Using Concatenated Convolutional Codes

Mahdi Zamani
Department of Electrical & Computer Eng.
University of Waterloo, Waterloo, Canada
Email: mzamani@cst.uwaterloo.ca

Farshad Lahouti
School of Electrical & Computer Eng.
University of Tehran, Tehran, Iran
Email: lahouti@ut.ac.ir

***Abstract*— A variable rate Slepian-Wolf (SW) code is constructed which is vital for wireless sensor network applications. A practical scheme to construct SW codes is to use the syndrome of a channel code as a compressed representation of a codeword in presence of a side information [3]. The proposed solution is based on an efficient and practical algorithm to compute the syndrome of rate-compatible convolutional codes (RCPC). By using this algorithm, there is no need to compute the syndrome of punctured version of the mother code for each puncturing matrix which is complex. Instead, the the syndrome of the punctured code is designed to be the punctured version of the syndrome of the mother code using the same pattern of puncturing. The algorithm is general for all convolutional codes in Z_q. The strategy is also generalized for parallel and serial concatenated convolutional codes.**

***Index Terms*— Slepian-Wolf theorem, Distributed source coding, Syndrome, Turbo code on rings.**

I. INTRODUCTION

Slepian-Wolf (SW) coding [1] and in general distributed source coding (DSC) has attracted considerable research interest in recent years. Slepian and Wolf [1] proved that separate compression and joint decompression of two correlated sources are as efficient as their joint compression and decompression.

The most important application of DSC is in wireless sensor networks (WSNs). In a WSN, about hundreds to thousands of energy limited sensor nodes are deployed in a region. Findings of nodes are highly correlated and exploiting this correlation for compression leads to energy saving and increasing the network lifetime. Since WSNs sense natural phenomena, the correlations among nodes change frequently. Thus, tracking these correlations and construction of a simple variable rate SW code are vital for efficient operation of WSNs.

The correlation between two sources can be modeled as a virtual channel referred to as the correlation channel. The input of the correlation channel is the first source X and its output is the second source Y referred to as side information (SI). If X and Y are two separate sources that are quantized and transmitted to a joint decoder, the correlation channel is discrete (X, Y and noise discrete). But when the joint decoder has access to continuous source Y or constructs a continuous side information Y at the decoder, e.g. from history of received data [2], the correlation channel is continuous (X discrete, Y and noise continuous). This channel has a capacity equal to the mutual information between the sources. If a channel code achieves this capacity, the mutual information between sources is extracted and only $H(X|Y)=H(X)-I(X;Y)$ bits/symbol are needed to reconstruct X and this is one of the corner points of the Slepian-Wolf boundary.

The proof of SW theorem is based on binning of one source and sending the index of the bin containing the source sequence. The joint decoder having the bin index finds the source sequence using the side information. A practical construction of Slepian-Wolf coding is also based on the binning approach. As constructed in [3], the source codebook X is partitioned into cosets of a channel code and the syndrome of the coset containing the source sequence S_X is transmitted to the joint decoder. At the joint decoder, the nearest sequence to its SI Y having the syndrome S_X is obtained as $\hat{X}$ using the channel decoder. Based on this approach, to compute the syndrome of sequences, a syndrome former (SF) must be constructed.

In SW coding based on channel codes, the more powerful the channel codes, the lower the decoding error. Then powerful channel codes, like parallel and serial concatenated convolutional codes and LDPC codes, whose performance are near the capacity of the correlation channel, are good candidates for the distributed source coding problem. Therefore, designing a simple syndrome former (SF) for such codes is essential. In general, the syndrome former of LDPC codes is more complex and energy consuming than that of the concatenated convolutional codes.

Another way of constructing a SW code is coding one source using a systematic code and sending only parity symbols of this codeword. At the decoder, the side information is considered as the systematic part of the codeword and the original data is found by decoding of this codeword. Prior studies indicate that the syndrome based scheme outperforms the parity based scheme [4].

As discussed, variable rate SW coding is essential in practical applications. A variable rate parity based SW code is presented in [5]. But, the work of Li and Alqamzi [6] has constructed a syndrome former (SF) for punctured convolutional codes. In this work, for each puncturing pattern, the corresponding generator matrix and a parity check matrix is derived. Subsequently, syndrome symbols are computed using this parity check matrix. This method is too complicated for many applications including WSNs, because for each compression rate or for each puncturing pattern, a new generator matrix should be derived using some complex

The material is based on research supported by the Iran Telecommunication Research Center (ITRC).

operations. In [6], only binary convolutional codes and discrete correlation channels are considered.

Since the correlated sources compressed using a SW code are, in general, continuous and, subsequently, quantized to symbols, to achieve improved performance a symbol-based turbo DSC scheme was suggested in [7].

In this paper, a variable rate SW code is constructed. For that, an efficient and general algorithm to produce the syndrome of concatenated convolutional codes over Z_q, a ring with q symbols, is proposed. The suggested algorithm is applicable to any concatenation scheme of convolutional codes and provides a flexible way to produce different coding rates. Also, it can be used in both discrete and continuous correlation channels. Using the presented strategy, to produce the syndrome of the punctured version of a code, one only needs to puncture the syndrome of the mother code using the same puncturing pattern. The proposed scheme provides a first solution for construction of high performance variable rate SW codes based on syndromes, with a small complexity.
The paper is organized as follows. Section II presents our method for constructing the syndrome former of convolutional codes. Section III generalizes the proposed algorithm to concatenated convolutional codes. Section IV presents the simulation results. Finally, section V concludes this article.

II. Variable Rate SW Code Construction Using RCPC Codes

Consider two correlated sources of data X_c and Y_c which are quantized to q-levels as X and Y (see Figure 1). We wish to compress X using a rate compatible punctured convolutional (RCPC) code with rate k/n, generator matrix $\mathbf{G}(D)=[g_{ij}(D)]_{k\times n}$ and constraint length v in Z_q, a ring with q symbols.

A rate k/n linear code, is in fact a k dimensional sub-space of the n-dimensional vector space. As a result, there always exists k independent (vector) columns of G. Without loss of generality we assume that these columns are the first k columns of G; Otherwise, one can easily swap the columns of the generator matrix. In such a setting, we consider the scenario where the puncturing is performed over the last n-k symbols of codewords.

Since the syndrome of X based on this code is the compressed representation of X, a syndrome former (SF) block is constructed to compute the syndrome. The decoder decompresses X according to its syndrome and its SI, i.e. Y or Y_C. All operations in this article are in Z_q.

A. At the source encoder

To compute the syndrome of X using a rate k/n convolutional code with constraint length v, where puncturing is performed over the last n-k symbols and the rate of punctured code is k/n_p, X is divided to n_p-symbol frames, X_1, X_2, …. Consider $\tilde{X}_i$ as the depunctured version of X_i, where in the deleted positions of X_i, dummy symbols are inserted (see Figure 2). Consider $\tilde{X}_i^s$ and $\tilde{X}_i^p$ as the first k symbols and the last n-k symbols of $\tilde{X}_i$, respectively.

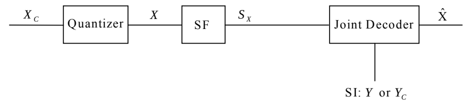

Figure 1. System model.

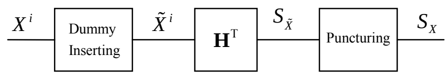

Figure 2. Syndrome former for RCPC codes.

To produce the syndrome of code, a parity check matrix is needed [3,4]. This parity check matrix is not unique and many parity check matrices can be found for this code. We use this degree of freedom in construction of the parity check matrix and find one which has the property that the syndrome of the punctured version of code is the punctured version of the syndrome of the code.

Consider $V_{\xi_{i-1}\xi_i}$ as the output corresponding to the branch between states ξ_{i-1} and ξ_i in the trellis of the code. $V^s_{\xi_{i-1}\xi_i}$ and $V^p_{\xi_{i-1}\xi_i}$ are the first k symbols and the last n-k symbols of $V_{\xi_{i-1}\xi_i}$, respectively. According to the assumed setting described in section II, for each $i\in\{0,1,...,q^v-1\}$, all members of the set $\left\{V^s_{\xi_i\xi_j}\right\}, j=0,1,...,q^k-1$ are different.

Now, we define n-k symbols as the syndrome of sequence X as follows and then we find its corresponding parity check matrix. We start from the first state of trellis which has q^k outgoing branches. There is one branch between the first state and state ξ_1, $V_{0\xi_1}$, where its first k symbols are equal to $\tilde{X}_1^s$. The syndrome of this block $\tilde{X}_1$ is defined as

$$S_1 \triangleq \tilde{X}_1^p + V^p_{0\xi_1}.(q-1) \tag{1}$$

For the next block, we start from state ξ_1 and find a branch, where $V^s_{\xi_1\xi_2}=\tilde{X}_1^s$, thus,

$$S_2 \triangleq \tilde{X}_2^p + V^p_{\xi_1\xi_2}.(q-1), \tag{2}$$

and so on. In the i'th block we have

$$S_i \triangleq \tilde{X}_i^p + V^p_{\xi_{i-1}\xi_i}.(q-1). \tag{3}$$

Using this procedure, each symbol of S_i is related to one and only one of the last n-k symbols of $\tilde{X}_i$ and puncturing of each of the last n-k symbols of $\tilde{X}_i$ means puncturing of its syndrome. As a result, we delete symbols of the syndromes corresponding to inserted dummy symbols in the depuncturing procedure as shown in Figure 2. The length of each resulting syndrome is n_p-k symbols.

To compute the syndrome using this approach, it is required to trace the trellis at the source encoder, which is a rather complex procedure. Note, that tracing the trellis for the case of n=2, k=1, and q=2 would be equivalent to the scheme suggested in [8] based on principal and complementary trellises. To devise a more efficient syndrome formation algorithm, we construct the parity check matrix that produces the desired syndromes, as described below.

At the first step, the above algorithm finds an output $V_{\xi_{i-1}\xi_i}$, where $V^s_{\xi_{i-1}\xi_i} = \tilde{X}_i^s$. $V_{\xi_{i-1}\xi_i}$ corresponds to an input $U_{\xi_{i-1}\xi_i}$, i.e.

$$\tilde{X}_i^s(D) = V^s_{\xi_{i-1}\xi_i}(D) = U_{\xi_{i-1}\xi_i}(D).[\mathbf{G}^1(D)]_{k\times k} \quad (4)$$

where $\mathbf{G}(D) = [g_{ij}(D)]_{k\times n}$ is divided into two separate matrices $[\mathbf{G}^1(D)]_{k\times k}$ and $[\mathbf{G}^2(D)]_{k\times(n-k)}$ as the first k and last n-k columns of $\mathbf{G}(D)$, respectively. Now, the desired syndrome is computed as

$$S_i(D) = U_{\xi_{i-1}\xi_i}(D).\mathbf{G}^2(D).(q-1) + \tilde{X}_i^p(D),\ i = 1,...,n \quad (5)$$

and we have

$$S_i(D) = \tilde{X}_i^s(D) \times (\mathbf{G}^1(D))^{-1}\mathbf{G}^2(D).(q-1) + \tilde{X}_i^p(D) \quad (6)$$

Note that according to the setting assumed in section II, $\mathbf{G}^1(D)$ is full-rank and therefore, invertible. Thus, the parity check matrix is given by

$$\mathbf{H}^T(D) = \begin{bmatrix} g'_{11}(D) & g'_{12}(D) & ... & g'_{1,n-k}(D) \\ g'_{21}(D) & g'_{22}(D) & ... & g'_{2,n-k}(D) \\ . & . & ... & . \\ g'_{k1}(D) & g'_{k2}(D) & ... & g'_{k,n-k}(D) \\ 1 & 0 & ... & 0 \\ 0 & 1 & ... & 0 \\ & .... & & \\ 0 & 0 & ... & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{G}'(D)_{k\times(n-k)} \\ \mathbf{I}_{(n-k)} \end{bmatrix}_{n\times(n-k)} \quad (7)$$

where

$$\mathbf{G}'(D) = \left(\mathbf{G}^1(D)\right)^{-1}.\mathbf{G}^2(D).(q-1) \quad (8)$$

and

$$S(D) = X(D)\mathbf{H}^T(D) \quad (9)$$

According to (7) and (9), syndromes of the mother code can be computed easily. And the syndrome of the punctured code is the punctured version of the syndrome of the mother code using the same puncturing pattern, i.e. syndrome elements at the position of the inserted dummy symbols are deleted. Thus, the source encoder only uses one parity check matrix to compute the syndrome.

B. At the source decoder

The source decoder uses a decoder of a convolutional code to find the nearest sequence to the side information with a syndrome equal to that of *X*. For that, the code trellis in the channel decoder, which uses either the Viterbi or the BCJR algorithm, are modified by the received syndrome, i.e. the codewords corresponding to the trellis branches are replaced with the following

$$\breve{V}_{\xi_i\xi_{i+1}} = V_{\xi_i\xi_{i+1}} + [0_{1\times k} \quad S_i]_{1\times n}. \quad (10)$$

Thus, the syndrome of the new codewords corresponding to the branches of the trellis is equal to that of *X*, *S*, because the first k bits of $\breve{V}_{\xi_i\xi_{i+1}}$ is equal to that of $V_{\xi_i\xi_{i+1}}$, thus $S(\breve{V}_{\xi_i\xi_{i+1}}) = V^p_{\xi_i\xi_{i+1}} + S_i + (q-1).V^p_{\xi_i\xi_{i+1}} = S_i$. Now, the decoder simply finds the nearest sequence of this trellis to the side information. This is equivalent to searching the coset with a syndrome equal to the received syndrome *S*.

III. Variable Rate SW Code Construction Using Concatenated Convolutional Codes

Traditional concatenated convolutional codes consist of two constituent convolutional codes $\mathbf{G}_1$ and $\mathbf{G}_2$ and one interleaver. To compute the syndrome of concatenated convolutional codes, the source sequence is converted to two sequences corresponding to two constituent code trellises and the syndrome is computed for each of them, separately, using the method discussed in section II.

At the decoder, the trellis of each constituent code should be modified according to (10). The metrics in either the BCJR algorithm or SOVA algorithm is computed using this modified trellis.

Below, we describe SW coding using two well-known concatenated convolutional codes: parallel concatenated convolutional codes (PCCC) and serial concatenated convolutional codes (SCCC).

A. SW Coding Using Parallel Concatenated Convolutional Codes (PCCC)

In a PCCC or turbo code the input symbols of the second constituent code is an interleaved version of the input symbols of the first constituent code. Consider a turbo code with rate $k/(2n-k)$ consisting of two systematic convolutional codes with rate *k/n*. Also, consider the case where puncturing is performed on parity symbols as shown in Figure 3(a) and the last rate is $k/(n_{p_1} + n_{p_2} - k)$. At first, the source data X is divided into $(n_{p_1} + n_{p_2} + k)$-symbol frames, X_1, X_2, … and $\tilde{X}_i$ is the depunctured version of X_i. Also, consider $\tilde{X}_i^s$, $\tilde{X}_i^{p_1}$ and $\tilde{X}_i^{p_2}$ as the first k symbols, n-k symbols after the first k symbols and the last n-k symbols of $\tilde{X}_i$ as shown in Figure 3(a). Two syndrome sequences are computed according to two convolutional codes using (7) and (9). The first syndrome, S^1, is computed for $[\tilde{X}_i^s \ \tilde{X}_i^{p_1}]$ and the second syndrome, S^2, for $[\Pi(\tilde{X}_i^s) \ \tilde{X}_i^{p_1}]$ as inputs of the syndrome former of two constituent codes, where Π is the interleaver function. At last, syndromes corresponding to the punctured symbols are deleted.

B. Serial Concatenated Convolutional Code (SCCC)

In a SCCC the input symbols of the second constituent code is an interleaved version of the output symbols of the first constituent code. Here we focus on SCCCs whose inner codes are systematic. A SCCC with rate *k/n* consists of an outer convolutional code $\mathbf{G}_2$ with rate k/n_1 and an inner convolutional code $\mathbf{G}_1$ with rate n_1/n. Also, consider that the puncturing is performed over the last n-n_1 symbols as shown in Figure 3(b) and the last rate is k/n_p. The depunctured version of each block of source $\tilde{X}_i$ of length n is divided into $\tilde{X}_i^s$, the first n_1 symbols, and $\tilde{X}_i^p$, the last n-n_1 symbols. Two syndrome sequences should be computed according to two convolutional codes using (7) and (9). One of them S^2 of length n_1-k is computed for $\Pi^{-1}(\tilde{X}_i^s)$ as inputs of the syndrome former of the outer constituent code and the other one S^1 of length n_p-n_1 is computed for $\tilde{X}_i$ as inputs of the syndrome former of the inner constituent code, respectively. At last, syndrome symbols corresponding to the punctured symbols are deleted.

IV. SIMULATION RESULTS

A. Convolutional Codes

Four two-source DSC schemes using a convolutional code are simulated. The generator matrix of the mother code is [*1,171/133,145/133,127/133*]. Puncturing among the parity bits is performed to achieve other compression rates. The dependency between two sources is considered as a BSC channel. Figure 4(a) shows the BER vs. crossover probability of the BSC model. The codes and the results are the same as those presented in [6]. However, as discussed in introduction, the complexity of the proposed scheme is substantially smaller. As mentioned, this is due to the fact that we do not reconstruct a syndrome former for each compression rate.

B. PCCC

SW coding using a non-binary PCCC is simulated. The generator matrix of the mother code of constituent codes in the PCCC is $\left[1, \frac{2+D}{1+3D}\right]$ on Z_8. The length of S-random interleaver is 10000. At the decoder, 20 iterations were used. The correlation model is $Y_C = X_C + n$, where n is a zero mean Gaussian random variable. A 8 level Llyod-Max quantizers is used. The results for compression rates of 1.5 and 2 are shown in Figure 4(b).

V. CONCLUSION

A simple and efficient algorithm to construct a variable rate SW code is proposed. In the proposed algorithm, there is no need to use another channel code or compute the syndrome for the punctured version of the channel code. The syndrome is computed simply using a specific parity check matrix and puncturing. The proposed method is generalized to turbo codes and serial concatenated convolutional codes. A method is proposed to compress two correlated sources using SW coding to achieve any point of the SW boundary. These methods are very practical for application of wireless sensor networks.

REFERENCES

[1] Slepian and J. K. Wolf, " Noiseless coding of correlated information sources," *IEEE Trans. of Inform. Theory*, pp. 471-480, Jul. 1973.

[2] J. Chou, D. Petrovic and K. Ramchandran, "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks," *IEEE INFOCOM*, San Franscisco, Ca Mar. 2003.

[3] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Trans. of Inform. Theory*, pp. 626-643, Mar. 2003.

[4] Z. Tu, J. Li (Tiffany) and R. S. Blum, "Compression of binary source with side information using parallel concatenated convolutional codes," *Proc. IEEE GLOBCOM*, Vol. 1, pp. 46-50, Nov. 2004.

[5] A. Aaron and B. Girod, "Compression with side information using turbo codes," *Proc. Of IEEE Data Compression Conference (DCC)*, pp. 252-261, Apr. 2002.

[6] J. Li and H. Alqamzi, "An optimal distributed and adaptive source coding strategy using rate-compatible punctured convolutional codes," *Proc. Of IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 3, pp. 685-688, Mar. 2005.

[7] M. Zamani and F. Lahouti, "Distributed source coding using symbol-based turbo codes," *Proc. 23rd Biennial Symp. Comm.*, Kingston, Ontario, Canada, May. 2006.

[8] 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)*, pp. 193-202, Mar. 2003.

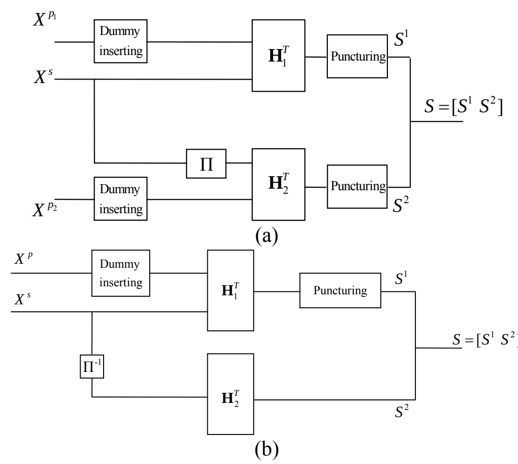

Figure 3. (a) The syndrome former of a PCCC. (b) The syndrome former of a SCCC.

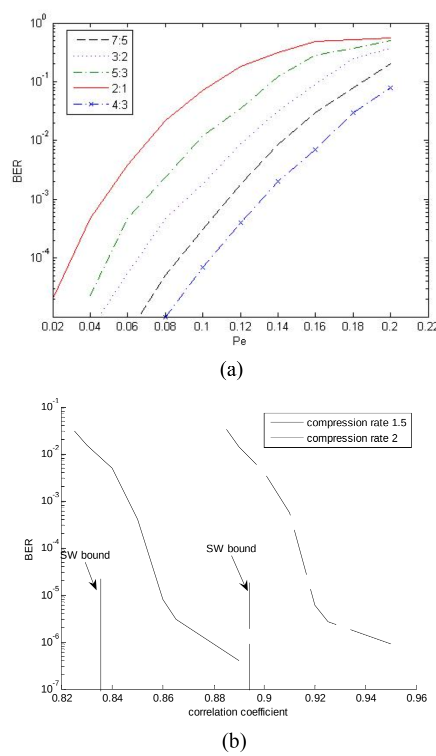

Figure 4. (a) BER vs. crossover probability of the correlation channel for RCPC codes. (b) BER vs. correlation coefficient of two sources.