Turbo and Trellis-Based Constructions for Source Coding with Side Information †

Jim Chou†, S. Sandeep Pradhan‡ and Kannan Ramchandran †
†EECS, University of California at Berkeley
‡EECS, University of Michigan at Ann Arbor

Abstract

We study the problem of rate-distortion efficient constructions for the problem of source coding with side information (SCSI), which has assumed heightened interest recently. While the Wyner-Ziv theorem from information theory has prescribed rate-distortion performance bounds for the SCSI problem, the gap between theory and practice has remained large. This work is aimed at reducing this gap. We propose two different frameworks based on a trellis construction and a turbo-based construction respectively. Our simulation results on the Gaussian SCSI problem reveal the promise of our proposed approaches: at 1 bit per sample, 0.5 bits/sample, 0.25 bits/sample and 0.125 bits/sample, our constructions attain performance within 1.3 dB, 1.1 dB, 0.85 dB and 0.5 dB respectively of the theoretical Wyner-Ziv rate-distortion bound.

1 Introduction

Consider the application of blind compression for dense sensor networks. In a sensor network, several nodes might be measuring correlated data, but each node has access to only its own measurement. To reduce the energy consumption of each node, it is desirable to have each node compress its measurement with respect to measurements from other nodes. This is possible by modeling the problem as a source coding with side information problem (SCSI), where the difference between the side information and source is modeled to be *i.i.d.* Gaussian. In the problem of Gaussian SCSI, a source is to be compressed given that there is correlated data available at the decoder and a distortion constraint between the reconstruction and the source. Wyner and Ziv established rate-distortion bounds for this problem in 1976 [1]. Until recently, however, there have been very few attempts at practical constructions. Recently, it has been realized that several applications such as blind compression for sensor networks can be well-modeled as source coding with side information and as a result it has become important to build practical codes that approach the SCSI performance bounds. There have been several attempts at constructing codes for SCSI [2, 3, 4] but the gap to the rate-distortion bound still remains relatively large.

It has been shown [2, 5] that in order to design good SCSI codes, one should partition a source codebook into channel codebooks. In this paper we propose two different code constructions for SCSI; one of the constructions is based on partitioning

†This work was supported in part by NSF-CCR-0219722

a trellis coded quantization codebook into trellis coded modulation codebooks and the other construction is based on partitioning trellis coded quantization codebooks into turbo coded modulation codebooks. The goal of the constructions is to bridge the gap between what is currently achievable and the rate distortion bound. Our simulations show that our constructions are powerful at several different encoding rates. Our best construction at encoding rates of 1 bit per sample, 0.5 bits/sample, 0.25 bits/sample and 0.125 bits/sample are within 1.3 dB, 1.1 dB, 0.85 dB and 0.5 dB respectively of the theoretical Wyner-Ziv rate-distortion bound.

2 Design Guidelines

In this section, we will derive design guidelines that will motivate our code constructions in the later sections.

The problem [1] of rate-distortion optimal lossy encoding of a source X with the side information Y available (losslessly) at the decoder is shown in Fig. 1(a). X

(a) (b)

Figure 1: (a) Source encoding with side information at the decoder. The encoder transmits at a rate greater than or equal to R bits/ source sample. (b) The *i.i.d.* Gaussian case.

and Y are correlated random variables with joint distribution $P(X,Y)$, such that the sequence pair $\{X_k, Y_k\}_{k=1}^{\infty}$ denotes independent realizations of the given random variables.

In this paper we will consider the case of $X = Y + Z$, where X, Y and Z are all *i.i.d.* Gaussian random variables with distributions, $P(X|Y) = \mathcal{N}(0,N)$, $P(Y) = \mathcal{N}(0,Q)$ and $P(Z) = \mathcal{N}(0,N)$ and the distortion measure is given as $d(x,\hat{x},y) = (x-\hat{x})^2$. The rate-distortion function, $R^*(D)$, for this system has been shown to be [1]:

$$R^*(D) = \left[\frac{1}{2}log_2\left(\frac{N}{D}\right)\right]^+. \tag{1}$$

To achieve the rate distortion bound, it can be shown [1] that U should be expressed as a linear combination of X and some noise $V \sim \mathcal{N}(0, \frac{D(N-D)}{N})$, $U = \alpha X + V$, where $\alpha = \frac{N-D}{N}$ is a scale factor. A depiction of this system is given in Fig. 1(b). From the figure, we can see that the decoder will have side information $Y = X - Z$ available to it which is correlated to X. The best estimate of X, represented by $\hat{X}$, will be given by $U + (1-\alpha)Y$. To find the U that the encoder used, the decoder should project U onto Y and find the closest U to the resulting projection, αY. In fact, αY can be expressed as U plus some noise, P:

$$\begin{aligned} \alpha Y &= \alpha(X - Z) \qquad (2)\\ &= U - \alpha Z - V = U + P \end{aligned}$$

where $P = -\alpha Z - V$. To successfully recover U then, the decoder must be able to correct for noise, P. From the above, we can see that a SCSI code must satisfy two constraints; at the encoder a codeword U must be found that satisfies a distortion constraint and at the decoder the codebook must correct for channel noise. One can construct a code to satisfy these two constraints by partitioning a source code into channel codes. Let us denote this partition by Λ_g/Λ_c where Λ_g is the source codebook and Λ_c is the channel codebook. The encoder will then specify the coset of Λ_c that the decoder should use as the channel codebook. In the following, we will determine the properties that the source codebook Λ_g and the channel codebooks Λ_c should satisfy.

From (2) if V is *i.i.d.* Gaussian, then P is also *i.i.d.* Gaussian because Z is *i.i.d.* Gaussian and a maximum likelihood decoder is optimal for channel decoding the side information. In practice, however, V will not be *i.i.d.* Gaussian, but we can still approximate P to be *i.i.d.* Gaussian for high encoding rates (i.e., $N >> D$), because in this case $\alpha = \frac{N-D}{N} \approx 1$ and therefore, $P \approx -Z$. We can define a probability of error to be the probability of decoding to a codeword that is different from the one used by the encoder. At high encoding rates, we can make use of the union bound to approximate this probability of error to be

$$\begin{aligned} P_e &\approx K_{min}(\Lambda_c) Q(d_{min}(\Lambda_c/2\sigma_p)) \\ &= K_{min}(\Lambda_c) Q((3\gamma_c(\Lambda_c)\gamma_g(\Lambda_g)/SNR_{norm})^{1/2}) \end{aligned} \tag{3}$$

This expression follows from [6], where we define $d_{min}(\Lambda_c)$ as the minimum distance between codewords in Λ_c, $K_{min}(\Lambda_c)$ as the number of nearest neighbors to any codepoint in Λ_c, $\gamma_g(\Lambda_g) = \frac{1/12}{G(\Lambda_g)}$ as the granular gain of Λ_g, $\gamma_c(\Lambda_c) = \frac{d^2_{min}(\Lambda_c)}{[V(\Lambda_c)]^{2/L}}$ as the coding gain of Λ_c and SNR_{norm} as a normalized-signal-to-noise ratio (see [6]). In defining the granular gain and the coding gain, we have used $G(\Lambda_g)$ to represent the normalized second moment of Λ_g and $V(\Lambda_c)$ to represent the volume of the Voronoi region of a codepoint. Note, that in the source coding literature, granular gain is a measure of the performance gain of using a vector quantizer as opposed to a scalar quantizer and in the channel coding literature, coding gain is defined to be a measure of packing efficiency with respect to a cubic constellation. From (4), we see that the probability of error can be decomposed into the two separate gains of coding gain and granular gain. Note, however, that this is only valid for high encoding rates. Nonetheless, it leads to the design rule that a strong source code needs to be found which can be partitioned into strong channel codes.

At low encoding rates, one can no longer de-couple the granular gain and the coding gain. Note that in this case $\alpha = \frac{N-D}{N} << 1$, and as a result the quantization error will significantly contribute to the variance of the noise:

$$\sigma_P^2 = \alpha^2 N + \alpha D \tag{4}$$

It therefore becomes important at low rates to use a lattice, Λ_g, that is a good source code to minimize the variance of the quantization error, D. One thing to note, however, is that the reconstruction of the source is given as $\hat{\mathbf{X}} = \mathbf{U} + (1-\alpha)\mathbf{Y}$ so that the average overall distortion, D^* will be:

$$
\begin{aligned}
D^* &= \frac{1}{L}E[||\mathbf{X}-\hat{\mathbf{X}}||^2] \\
&= \frac{1}{L}E[||\mathbf{X}-(\mathbf{U}+(1-\alpha)\mathbf{Y})||^2] \\
&= \frac{1}{L}(E[||\alpha\mathbf{X}-\mathbf{U}||^2]+E[(1-\alpha)^2||\mathbf{X}-\mathbf{Y}||^2]) \\
&= \alpha D+(1-\alpha)^2 N
\end{aligned} \tag{5}
$$

where the last equality follows from the independence of the quantization error and the correlation noise. Thus, when $\alpha \approx 0$, the distortion will be approximately N and it is sufficient to reconstruct the source as the side information itself. This implies that at very low transmission rates both the coding gain and the granular gain become moot.

3 Trellis Construction

From the previous section, we showed that a good SCSI code consists of a source code that achieves large granular gain and can be partitioned into cosets of a channel code that achieves large coding gain. Furthermore, in order for the code to be practical, there should be efficient encoding and decoding algorithms associated with the code. To achieve these goals, we will focus our attention on the special case of the source code and its partitions being trellis codes. We will start by showing how a trellis code can be partitioned and then once the trellis partition is formed we will propose how it can be used for a SCSI encoder and a SCSI decoder.

We start by considering trellis codes that can be represented as generalized coset codes [7]. To define generalized coset codes, we first need to introduce a signal space code, $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})$ (see [7]), as a label alphabet $\mathcal{A}$, an index set I, a label code $\mathcal{C}$ equal to any subset of the sequence space $\mathcal{A}^{\mathcal{I}}$, a partition Λ/Λ' of a signal set Λ into subsets, and a one-to-one label map m: $\mathcal{A} \to \Lambda/\Lambda'$. Quoting the definition given in [7], the code $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})$ is the disjoint union

$$
\mathbb{C}(\Lambda/\Lambda';\mathcal{C}) = \bigcup_{c\in\mathcal{C}} \mathbf{m}(\mathbf{c}) \tag{6}
$$

of the subset sequences $\mathbf{m}(\mathbf{c}) = \{m(c_k), k \in I\}$, $c \in \mathcal{C}$; i.e., $\mathbf{m}(\mathbf{c})$ is the subset sequence selected by the label sequence $\mathbf{c} \in \mathcal{C}$ via the label map $\mathbf{m}$. A generalized coset code, then, is the set of all sequences $\{\lambda \in \mathbf{m}(\mathbf{c}), \mathbf{c} \in \mathcal{C}\}$, where $\mathcal{C}$ is a subgroup of $\mathcal{A}^I$, I is some index set, and $\mathbf{m} : \mathcal{A}^I \to (\Lambda/\Lambda')^I$ is the sequence extension of the mapping $m : \mathcal{A} \to \Lambda/\Lambda'$. An example of a generalized coset code is given in Fig. 2(a). In the example, we take $\Lambda/\Lambda' = \mathbb{Z}/4\mathbb{Z}$ and $\mathcal{C}$ to be a rate-1/2 convolutional code. The input data, $\mathbf{f}$, will feed the convolutional code and determine the subset sequence, $\mathbf{m}(\mathbf{c})$, and the "other data", $\mathbf{g}$, will choose the signal sequence. The union of all possible sequences forms the generalized coset code. Note in the example, that because Λ and Λ' contain an infinite set of points, the input sequence, $\mathbf{g}$, will

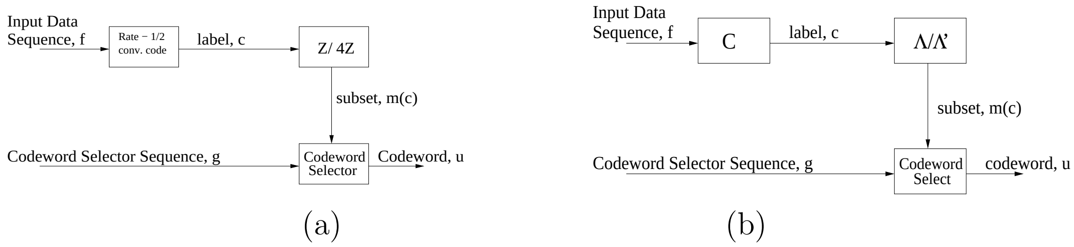


Figure 2: (a) Generalized coset code example. $\Lambda/\Lambda' = \mathbb{Z}/4\mathbb{Z}$. The code used to select the subset sequence is a rate-1/2 convolutional code. (b) General representation of a generalized coset code.

consist of an infinite set of parallel bit sequences to index the signal sequence. A general representation of a generalized coset code is given in Fig. 2(b). The benefit of choosing the trellis code to be a generalized coset code, is that generalized coset codes are geometrically uniform and also lead to geometrically uniform partitions. For a geometrically uniform partition, the distance profile of each coset is the same. Furthermore, the Voronoi regions of all signal sequences, λ associated with $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})$ will be of the same shape. This implies that if we choose a trellis code based on $\mathbb{C}(\Lambda/\Lambda';\mathcal{C}')$ as a channel code then the probability of error will not depend on the codeword. Moreover, if we choose a trellis code based on $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})$ as a source code then the performance should be the same across all codewords. In the following, we will construct a SCSI code that consists of a trellis source code that partitions into cosets of a trellis channel code.

3.1 Construction

Using the above notation, let $\mathcal{C}$ be a rate-$\frac{n}{m}$ convolutional code and let $\mathcal{C}'$ be a rate-$\frac{k}{m}$ convolutional code that is constructed by concatenating a rate-$\frac{k}{n}$ convolutional code with $\mathcal{C}$ (see Fig. 3). Given that modulo addition is the group operation defined over the elements of $\mathcal{C}$, then $\mathcal{C}'$ is a normal subgroup of $\mathcal{C}$ and the signal space code defined by $\mathbb{C}(\Lambda/\Lambda';\mathcal{C}')$ is a subset of $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})$. A coset of $\mathbb{C}(\Lambda/\Lambda';\mathcal{C}')$ in $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})$ can be defined as $\mathbb{C}(\Lambda/\Lambda';\mathcal{C}'+\mathbf{a})$ for any $\mathbf{a} \in \mathcal{C}$ and using techniques of [7], it can be shown that $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})/\mathbb{C}(\Lambda/\Lambda';\mathcal{C}')$ forms a geometrically uniform partition. We can therefore define a SCSI code by the partition, $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})/\mathbb{C}(\Lambda/\Lambda';\mathcal{C}')$ where the source is trellis encoded using $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})$ and the coset index is output to the decoder. The decoder will use the bits from the encoder to index a coset of $\mathbb{C}(\Lambda/\Lambda';\mathcal{C}')$ to trellis decode the side information. We can see that $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})/\mathbb{C}(\Lambda/\Lambda';\mathcal{C}')$ is formed through the partition $\mathcal{C}/\mathcal{C}'$. To partition $\mathcal{C}$ will involve partitioning the 2^{nL} sequence space (where L is the dimension of the sequence) that constitutes the output of the rate-$\frac{k}{n}$ convolutional code. The end result will be a rate-$\frac{n}{m}$ convolutional code that partitions into disjoint classes of rate-$\frac{k}{m}$ convolutional codes. The partition can be implemented in practice by having the encoder send the syndrome corresponding to the rate-$\frac{k}{n}$ convolutional code [2]. The syndrome will completely specify the coset of $\mathbb{C}(\Lambda/\Lambda';\mathcal{C}')$ that the decoder should use for trellis decoding [2].

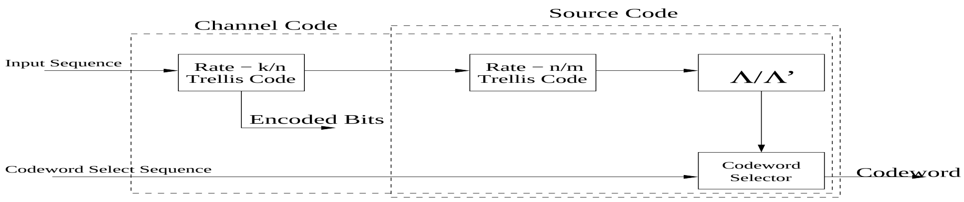


Figure 3: Trellis construction: a rate - k/m trellis is embedded within a rate n/m trellis, where m > n > k.

3.1.1 Encoder

The encoder will search the input sequence to $\mathcal{C}$ and the input sequence for the signal point selector to find the closest codeword, $\mathbf{U}$, in the trellis code associated with $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})$ to $\alpha\mathbf{X}$, where $\mathbf{X}$ represents the source sequence. If we represent the input sequence to $\mathcal{C}$ as a polynomial $\mathbf{d}(D)$, then the encoder will transmit $\mathbf{m}(\mathbf{D}) = \mathbf{H}(D)\mathbf{d}(D)$ to the decoder, where $\mathbf{H}(D)$ represents the parity check matrix for the rate-$\frac{k}{n}$ convolutional code. Given that Λ and Λ' have dimension M, then the rate of encoding is given as $\frac{n-k}{M}$ bits per source sample.

3.1.2 Decoder

The decoder will receive $\mathbf{m}(D)$ from the encoder and will also have access to some side information that is correlated to the source, $\mathbf{Y} = \mathbf{X} - \mathbf{Z}$, where $\mathbf{Z}$ is *i.i.d.* Gaussian. The decoder will first scale $\mathbf{Y}$ by α and then index a coset of the signal space code, $\mathbb{C}(\Lambda/\Lambda';\mathcal{C}')$ using $\mathbf{m}(D)$. The input sequences to $\mathcal{C}'$ along with the input sequences to the signal point selector are then searched for the set of inputs that minimize the squared error between the codeword and $\alpha\mathbf{Y}$. A linear combination of the decoded codeword and the side information, $\mathbf{U} + (1-\alpha)\mathbf{Y}$ is then used as the reconstruction of the source.

3.2 Turbo codebooks

We can further strengthen the SCSI constructions by incorporating turbo trellis coded modulation (TTCM) into the channel codebooks. In particular, we propose to append an interleaver in between the rate-$\frac{k}{n}$ convolutional code and the rate-$\frac{n}{m}$ convolutional code of the trellis-based construction. The trellis code used for encoding the source will then be based on the signal space code, $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})$, where $\mathcal{C}$ is the rate-$\frac{n}{m}$ convolutional code. The trellis code used for decoding the side information will be based on the signal space code, $\mathbb{C}(\Lambda/\Lambda';\mathcal{C}' + \mathbf{a})$, where $\mathcal{C}'$ is the concatenation of the rate-$\frac{k}{n}$ convolutional code with an interleaver followed by the rate-$\frac{n}{m}$ convolutional code and $\mathbf{a} \in \mathcal{C}$. The encoding and decoding operations are given in the following subsections.

3.2.1 Encoder

To encode, the input sequence to $\mathcal{C}$ along with the input sequence to the signal point selector is searched to find the set of inputs that minimize the squared error between the resulting codeword and the scaled source sequence, $\alpha\mathbf{X}$. After the input sequence is found, it is fed through a de-interleaver to create a sequence, $\mathbf{d}'(D)$. The encoder will multiply $\mathbf{d}'(D)$ by the parity check matrix of the rate-$\frac{k}{n}$ convolutional code and send the result, $\mathbf{m}(D) = \mathbf{H}(D)\mathbf{d}'(D)$, to the decoder.

3.2.2 Decoder

The decoder will receive $\mathbf{m}(D)$ from the encoder and will have access to some side information, $\mathbf{Y} = \mathbf{X} - \mathbf{Z}$. The decoder will index a coset of $\mathbb{C}(\Lambda/\Lambda';\mathcal{C}')$ and then use the corresponding trellis code to try and find the closest codeword to $\alpha\mathbf{Y}$. The Viterbi algorithm is not suitable for this task as a result of the excessive number of paths that need to be searched. Instead, we propose to use a suboptimal turbo decoding algorithm to find the closest codeword, $\mathbf{U}$ to $\alpha\mathbf{Y}$. A block diagram of the decoder is given in Fig. 4. In the figure, the decoder starts by decoding $\alpha\mathbf{Y}$ with respect to $\mathbb{C}(\Lambda/\Lambda';\mathcal{C})$ by using a modified BCJR algorithm [8]. A brief description of the modified BCJR algorithm is as follows. If we let $Y'_k = \alpha Y_k$ represent the k^{th} scaled sample of the side information, then the branch transition probability at stage k of the trellis is given by

$$\begin{aligned} \gamma_i(Y'_k, M', M) &= P(d_k = i, Y'_k, T_k = M | T_{k-1} = M') \\ &= P(Y'_k | d_k = i, T_k = M, T_{k-1} = M') q(d_k = i | T_k = M, T_{k-1} = M') P(T_k | T_{k-1}) \end{aligned} \tag{7}$$

where d_k is the n-bit input to the rate-$\frac{n}{m}$ convolutional code and T_k represents the trellis state at stage k. Now, $q(d_k = i | T_k = M, T_{k-1} = M')$ will be the a priori information that is to be passed from the rate-$\frac{k}{n}$ decoder and $P(Y'_k | d_k = i, T_k = M, T_{k-1} = M')$ can be approximated as:

$$\begin{aligned} P(Y'_k | d_k = i, T_k = M, T_{k-1} = M') &= P(Y'_k | u) \\ &= P(u - V_k - \alpha Z_k) \\ &= \frac{1}{\sqrt{2\pi}\sigma_P} exp\left(-\frac{(Y' - u)^2}{2\sigma_p^2}\right) \end{aligned} \tag{8}$$

where $Z_k \sim \mathcal{N}(0, N)$ is the correlation noise, V_k is the quantization error introduced by the encoder and $\sigma_P^2 = \alpha N$.

Once the a priori probabilities are calculated from the rate-$\frac{n}{m}$ code, these probabilities are fed through an interleaver and to the decoder for the rate-$\frac{k}{n}$ code. The decoder for the rate-$\frac{k}{n}$ code is a modified BCJR algorithm that takes as input the a priori probabilities from the BCJR decoder of the rate-$\frac{n}{m}$ code and also $\mathbf{m}(D)$. It will then calculate a priori probabilities for $\mathbf{m}(D) + \mathbf{c}'(D)$, where $\mathbf{c}'(D)$ is the polynomial representation of a codeword in the rate-$\frac{k}{n}$ convolutional code and $\mathbf{m}(D)$ is the polynomial representation of the index transmitted from the encoder. These a priori

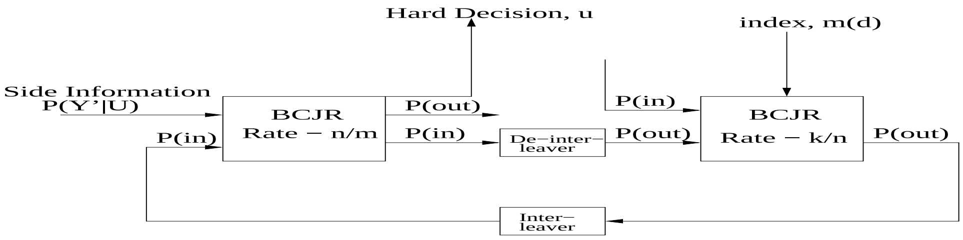


Figure 4: Turbo decoder for SCSI system: The BCJR decoder for the rate-$\frac{n}{m}$ code receives its a priori input probabilities from the rate-$\frac{k}{n}$ decoder and the BCJR decoder for the rate-$\frac{k}{n}$ code receives its a priori input probabilities from the rate-$\frac{n}{m}$ decoder.

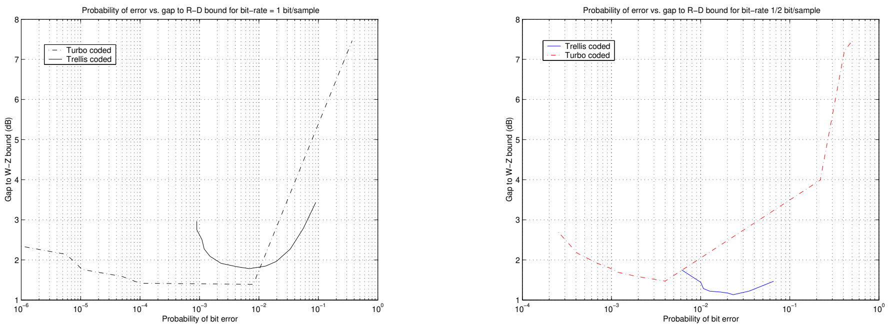
probabilities will be passed through an interleaver and used by the BCJR decoder for the rate-$\frac{n}{m}$ code.

The above process continues for a specified number of iterations and then the rate-$\frac{n}{m}$ decoder will output an estimate of the codeword, $\mathbf{U}$, that was used by the encoder.

The design guidelines used in designing the turbo TCM channel code are similar to the ones proposed in [9]. In that paper, it was conjectured that the probability of error is inversely proportional to the interleaver length.

4 Results

We simulated the SCSI code constructions for many different encoding rates. The first simulation that we ran was at an encoding rate of 1 bit per sample. The results are shown in Fig. 5(a). From the graph, we can see two different curves; one



(a) (b)

Figure 5: (a) Simulation results at 1 bit/sample. The trellis construction is 1.8 dB away from the R-D bound and the turbo construction is 1.3 dB away from the R-D bound. (b) Simulation results for 0.5 bits/sample.

curve corresponds to a trellis-based construction and the other curve corresponds to a turbo-based construction. The trellis-based construction uses the trellis partition

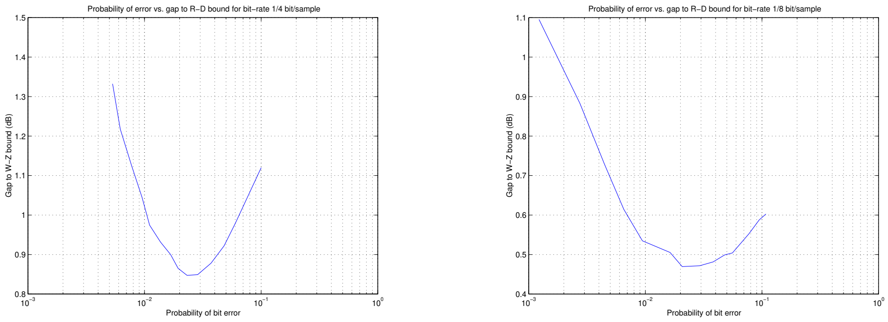
$\mathbb{C}(\mathbb{Z}/4\mathbb{Z};\mathcal{C})/\mathbb{C}(\mathbb{Z}/4\mathbb{Z};\mathcal{C}')$, where the parameters for $\mathcal{C}_1$ and $\mathcal{C}_2$ are given in Table 1. We let $\mathcal{C}_1$ and $\mathcal{C}_2$ represent the codes corresponding to the rate-$\frac{k}{n}$ and rate-$\frac{n}{m}$ convolutional codes, respectively (see Fig. 3). Furthermore, in the table, we list the parity check polynomials corresponding to $\mathcal{C}_j$ as $h_i(\mathcal{C}_j)$ where $\mathbf{H} = [h_n(\mathcal{C}_j) \;\; h_{n-1}(\mathcal{C}_j) \;\; \ldots \;\; h_0(\mathcal{C}_j)]$ is the parity check matrix for $\mathcal{C}_j$ and $h_i(.)$ is represented in octal notation. From

Rate	$\eta(\mathcal{C}_1)$	$h_2(\mathcal{C}_1)$	$h_1(\mathcal{C}_1)$	$h_0(\mathcal{C}_1)$	$\eta(\mathcal{C}_2)$	$h_2(\mathcal{C}_2)$	$h_1(\mathcal{C}_2)$	$h_0(\mathcal{C}_2)$
1	4	–	10	23	4	–	10	23
1	2	–	02	05	2	–	02	05
0.5	4	16	04	23	4	16	04	23
0.5	2	16	02	05	2	–	02	05

Table 1: Codes that were used in the simulations for 1 bit/sample and 0.5 bits/sample.

the figure, we can see that the trellis-based construction is about 1.8 dB away from the rate-distortion bound. The other construction is based on the turbo construction of Section 3.2. This construction is based on the convolutional codes $\mathcal{C}_1$ and $\mathcal{C}_2$ with their parameters given in row 2 of Table 1 and an interleaver length of 10,000 with a maximum of 18 iterations. It is about 1.3 dB away from the rate-distortion bound. Note, that we plotted the gap to the rate-distortion bound as a function of the probability of decoding error and it is actually beneficial to have a non-zero probability of error. We have defined the probability of error to be the probability that the decoder decodes to a codeword that is different than the one chosen by the encoder. The reason that a non-zero probability of error results in a minimal gap to the rate-distortion bound is that the overall distortion is a weighted average of the distortion that results when an error occurs and when an error does not occur. This average turns out to have a minimum at a non-zero probability of error. An analog to the probability of error in pure source coding is the probability of overload.

Next, we simulated an encoding rate of 0.5 bits/sample. For this encoding rate, we again considered two different constructions; one was a trellis-based construction and the other was a turbo-based construction. The trellis-based construction uses the trellis partition $\mathbb{C}(\mathbb{Z}^2/2R\mathbb{Z}^2;\mathcal{C})/\mathbb{C}(\mathbb{Z}^2/2R\mathbb{Z}^2;\mathcal{C}')$, where the parameters for $\mathcal{C}_1$ and $\mathcal{C}_2$ are given in row 3 of Table 1. The turbo-based construction uses the convolutional codes $\mathcal{C}_1$ and $\mathcal{C}_2$ that have parameters given in row 4 of Table 1. The performance of the two constructions is plotted in Fig. 5(b). From the figure, we can see that the trellis construction actually outperforms the turbo construction. The trellis construction is within 1.1 dB of the rate distortion bound and the turbo construction is within 1.5 dB of the rate distortion bound. One reason for this is that the turbo construction uses a weak source code (constraint length 2) in order to achieve good channel coding performance, *but at low encoding rates the quantization error significantly contributes to the overall noise.* Furthermore, the overall distortion becomes increasingly dependent on the side information (see Sec. 2) as the rates are lowered. This can be explained by noting that the reconstruction of the source is a weighted



(a) (b)

Figure 6: (a) Simulation results for trellis-based SCSI at 0.25 bits/sample. (b) Simulation results for 0.125 bits/sample.

combination of the side information and the decoded codeword, and as the encoding rates decrease the weighting of the side information increases. At the extreme case of 0 encoding rate, the reconstruction will be exactly the side information. For example, we simulated our trellis construction for encoding rates of 0.25 and 0.125 bits/sample. At these rates, the gap to the rate-distortion bound is 0.8 and 0.5 dB respectively. Plots of the performance are shown in Fig. 6(a) and Fig. 6(b).

The convolutional codes $\mathcal{C}_1$ and $\mathcal{C}_2$ that were used in the 0.25 bit/sample and 0.125 bit/sample simulations are given in Table 2.

Rate	$\eta(\mathcal{C}_1)$	$h_1(\mathcal{C}_1)$	$h_0(\mathcal{C}_1)$	$\eta(\mathcal{C}_2)$	$h_3(\mathcal{C}_2)$	$h_2(\mathcal{C}_2)$	$h_1(\mathcal{C}_2)$	$h_0(\mathcal{C}_2)$
0.25	2	02	05	4	01	06	01	25
0.125	2	02	05	4	30	14	02	61

Table 2: Code that was used in the simulation for 0.25 and 0.125 bits/sample

References

[1] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. on Inform. Theory*, vol. IT-22, pp. 1–10, January 1976.

[2] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes: Design and construction," *Proceedings of the Data Comression Conference (DCC)*, March 1999.

[3] X. Wang and M. T. Orchard, "Design of trellis codes for source coding with side information," *Proc. IEEE DCC*, March 2001.

[4] A. Aaron and B. Girod, "Compression with side information using turbo codes," *Proceedings of IEEE DCC*, March 2002.

[5] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Transactions on Information Theory*, vol. 48, pp. 1250–1276, June 2002.

[6] M. V. Eyuboglu and G. D. Forney, "Lattice and trellis quantization with lattice and trellis bounded codebooks – high-rate theory for memoryless sources," *IEEE Transactions on Information Theory*, vol. 39, pp. 46–59, January 1993.

[7] G. D. Forney, "Geometrically uniform codes," *IEEE Transactions on Information Theory*, vol. 37, pp. 1241–1260, September 1991.

[8] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. on Information Theory*, pp. 284–287, Mar 1974.

[9] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Trans. on Information Theory*, vol. 44, pp. 909–926, May 1998.