Design and Applications of Turbo Source Codes

Patrick Mitran

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

August 2002

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.

Your file Votre référence
ISBN: 0-612-85892-8
Our file Notre référence
ISBN: 0-612-85892-8

1,2	2,3	3,1	4,6	5,9	6,4	7,8	8,7	9,5	0,0
7,4	4,2	2,7	0,9	6,1	5,8	8,5	9,0	3,3	1,6
5,1	1,4	4,5	6,7	0,8	8,0	9,3	2,2	7,6	3,9
0,7	7,1	1,0	3,8	8,3	9,2	4,4	5,6	2,9	6,5
3,5	5,7	7,3	8,2	9,4	1,1	0,6	4,9	6,0	2,8
2,0	0,5	5,2	9,1	7,7	3,6	1,9	6,3	4,8	8,4
4,3	3,0	0,4	5,5	2,6	7,9	6,2	1,8	8,1	9,7
8,9	9,8	6,6	2,4	3,2	0,3	5,0	7,5	1,7	4,1
6,8	8,6	9,9	7,0	1,5	4,7	2,1	3,4	0,2	5,3
9,6	6,9	8,8	1,3	4,0	2,5	3,7	0,1	5,4	7,2

A pair of MOLS of order 10. (Cover page of Scientific American, Nov. 1959)

Abstract

The mathematical theory of communication has grown considerably since its inception 50 years ago. There are many problems that have been solved from the information theoretic perspective, yet remain open from the coding point of view. In particular, it is known what the optimum achievable performance of a certain system is, yet no practical coding scheme that achieves the optimal performance is known. This thesis is concerned with two such source coding problems: the Slepian-Wolf problem and the Wyner-Ziv problem. This thesis also investigates the related source coding problems of data compression and noise robust data compression.

A unified framework, based on the parallel concatenation of trellis structured codes (turbo codes) is applied and shown to perform well in all cases. This represents a break with traditional source coding techniques in that the code is fixed-length to fixed-length. As such, it is a probabilistic coding technique. An explicit joint design of the parallel concatenated codes, based on conditions rooted in information theory, is presented. The codes thus designed are intimately related to Latin squares and are therefore named Latin Square Based Codes. As opposed to the vast majority of the existing literature on turbo codes, these codes perform data compression and are designed jointly. Furthermore, they are non-binary, non-linear, non-systematic and non-symmetric.

In all the above cases, near Shannon limit performance is observed. For data compression as applied to binary memoryless sources, the scheme achieves reliable communication at a rate only 7.5% above the entropy of the source. A similar result is shown for the Slepian-Wolf problem. Noise robust compression is shown to be as close as 1.11 dB from capacity for AWGN channels while coding for the Wyner-Ziv problem is as close as 1.1 dB from the rate-distortion function.

Sommaire

La théorie mathématique des communications s'est développée considérablement depuis son commencement il y a 50 ans. Il y a beaucoup de problèmes qui ont été résolus de la perspective théorétique de l'information mais qui demeurent sans solution du point de vue du codage. En particulier, on connaît la performance optimale qu'un certain système peut réaliser, pourtant aucun code pratique qui atteint cette performance n'est connu. Cette thèse aborde deux tels problèmes de codage de source: le problème de Slepian-Wolf et le problème de Wyner-Ziv. Cette thèse étudie également deux problèmes relatifs au codage de source: la compression de données ansi que la compression de données robuste au bruit.

Un cadre unifié, basé sur la concaténation parallèle des codes structurés par treillis (codes turbo), est appliqué et il est demontré que ceci fonctionne bien dans tous les cas. La technique proposée se dissocie des techniques traditionnelles de code source parce que le code construit ainsi est de longueur-constante à longueur-constante. Il s'agit d'une technique probabiliste de codage. Une conception commune et explicite des codes enchaînés en parallèle, basée sur des conditions issues de la théorie de l'information, est présentée. Les codes ainsi conçus sont intimement liés aux carrés latins et sont donc appelés des codes basés sur carrés latins. Par opposition à la grande majorité de la littérature existante à propos des codes turbo, ces codes exécutent la compression de données et sont conçus conjointement. En outre, ils sont non binaires, non linéaires, non systématiques et dissymétriques.

Dans tous les cas ci-dessus, une performance près de la limite de Shannon est observée. Pour la compression de données pour des sources binaires sans mémoire, le système peut réaliser une communication fiable à un taux seulement de 7,5% au-dessus de l'entropie de la source. Un résultat semblable est obtenu pour le problème de Slepian-Wolf. La compression robuste au bruit est à 1,11 dB près de la capacité pour des canaux de bruit blanc gaussien superposé, tandis que le codage pour le problème de Wyner-Ziv est aussi près que 1,1 dB de la borne du taux de distorsion.

Acknowledgments

I would like to thank my supervisor, Dr. Jan Bajcsy, for introducing me to turbo codes and the Slepian-Wolf problem and without whom this work could not be possible. I would also like to thank my brother, Marcel Mitran, and my parents for their encouragement and support as well as my friend Rudolf who was always willing to take my mind off research (à l'aventure!).

Also, this thesis could not have been produced without the help of many people. Charif Beainy, Mark Klein, Naveen Mysore, Paxton Smith, Alexander Wyglinski and Raymond Yim all deserve thanks for technical help and proof-reading in the production of this thesis. I must also express my gratitude to Dr. Peter Kabal for his expertise in Latex and providing me with the Lloyd-Max quantization levels employed in Chapter 4 and Dr. Peter Caines for insightful discussions on state spaces. I am grateful for the generous financial support offered by the National Science and Engineering Research Council (Canada) and the Canadian Institute of Telecommunications Research.

Contents

List of Figures

List of Tables

List of Terms

AWGN Additive White Gaussian Noise. See [1].

BCJR Algorithm The Bahl, Cocke, Jelinek and Raviv Algorithm [2]. As opposed to the Viterbi Algorithm which computes the most likely sequence of symbols, the BCJR Algorithm computes the probabilities of the individual message symbols.

BPSK Binary Phase Shift Keying. See [1].

BSC Binary Symmetric Channel. A noise model commonly employed for binary channels. This channel is compeletly characterized by the crossover probability $P[X \neq Y] = \epsilon$.

CSNR Correlation Signal to Noise Ratio. A measure of the correlation between two Gaussian random variables. Let X and U be independent Gaussian random variables with variances σ_X^2 and σ_U^2 respectively. If $Y = X + U$, then the CSNR between X and Y is calculated as $10 \log_{10} \sigma_X^2 / \sigma_U^2$.

FSM Finite State Machine. This term is referenced in the context of FSM encoders.

i.i.d. Independent and identically distributed.

LDPC Low Density Parity Check Code [3]. A very powerful channel code based on sparse parity check matrices.

log The logarithm base 2 function.

LP Linear Prediction. See [4].

OPTA and **OPTA Gap** The Optimum Performance Theoretically Achievable and the difference between the measured performance of a code and the optimum performance theoretically achievable.

SNR Signal to Noise Ratio. When measured in decibels, this is expressed as $10 \log_{10} P/N$ where P is the signal power and N is the noise power.

VA Viterbi Algorithm [5]. An efficient algorithm to perform maximum likelihood sequence estimation.

VLC Variable Length Code.

Chapter 1

Introduction

Information theory [6, 7] has grown considerably since its inception 50 years ago. There are many problems that have been solved from the information theoretic perspective in the sense that the optimum achievable performance is known. Nevertheless, many of these remain open in the sense that there is no known coding technique with reasonable complexity that comes close to this performance.

This chapter first motivates a few such problems that will be dealt with in this thesis. A more formal framework to describe these problems is then presented. The chapter is concluded with a discussion on the original contribution and organization of this thesis. A more detailed analysis of the problems may be found in Chapter 2.

1.1 Practical Motivation

Modern technology has led to an explosion of information that must be processed, stored and communicated. Consider, for example, medical images and facsimile which require large storage capacity [8, 9]. Generally, only a few of these images are ever needed at once. Since most of these images contain large amounts of redundancy, storage costs can be reduced by compressing the data (i.e. eliminating the redundancy). When an image is needed, it can be decompressed at that time. This process is called source coding and a source code is said to be optimal if it can eliminate all the source redundancy.

In practice, information must often be transmitted over various, usually noisy, channels. Traditionally, this has been accomplished in a two step approach. First, the data is source encoded to minimize communication time, storage space and bandwidth. This is followed

by a channel encoding that introduces a controlled amount of redundancy to protect the data against channel errors. Without the latter, reliable communication is impossible. For transmission over noisy channels, it is not uncommon to use source encoders that are suboptimal. To illustrate, LP speech vocoders produce outputs that still contain residual redundancy [4, 10] while the Huffman code of a two-tone Group 3 facsimile is fixed and cannot be "fine tuned" to the statistics of the image in question [9]. Since the performance in terms of error protection against the noisy channel can be improved by this residual redundancy, methods that achieve the improved performance are desirable.

Although digital communication is more pervasive today than ever, most digital communication systems ignore the fact that existing analog channels are still physically present. If the analog transmission is included as extra "noisy side-information" available at the digital decoder, it is intuitively clear that better performance can be expected in terms of the minimum rate needed by the digital system for reliable communication.

Generally, most real sources of information are analog. With digital communication, it is impossible to encode such a source with perfect fidelity. It is usually desired to transmit the information subject to some acceptable loss of fidelity. Consider a wireless sensor array where the density of nodes is high (dozens per square meter) and communication bandwidth is limited [11]. There, for spatially close sensors, the observations will be nearly identical and it would be economical to take advantage of this correlation. Furthermore, this must be accomplished while minimizing some given distortion measure.

1.2 Problem Framework

Formally, consider a variation on the traditional communication system as illustrated in Fig. 1.1. Here, a variety of scenarios may be considered by the various settings of switches s_1 and s_2. In the case that both s_1 and s_2 are open (i.e. $s_1 = 0$ and $s_2 = 0$), one obtains the traditional point-to-point communication system, where a single source (denoted as X), is first encoded, then transmitted over (a possibly noisy) channel and finally decoded to produce $\hat{X}$, an approximation to X.

If the channel is noiseless, the system reduces to traditional data compression. For a uniform source and a noisy channel, such as either the Binary Symmetric Channel (BSC) or the Additive White Gaussian Noise (AWGN) channel, the system involves channel coding. For a source with redundancy, both the source and channel coding may be combined and

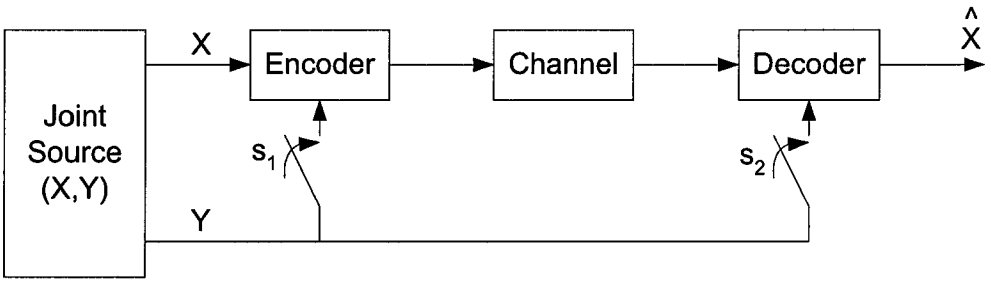

Fig. 1.1 A variation on the traditional communications model.

is commonly referred to as joint source-channel coding. In some cases, if the source has sufficient redundancy, it may even be possible to construct a joint source-channel code that produces fewer output symbols than input symbols (of the same alphabet). To distinguish such situations from the usual convention of joint source-channel coding where the encoder introduces additional redundancy, the former is denoted as noise robust compression (this emphasizes the fact that the encoder compresses the data, yet the output of the encoder is still robust against channel errors).

Consider a final case where switch s_1 is open, switch s_2 is closed, the channel is noiseless and perfect reconstruction of X (with probability of error arbitrarily small) is desired. This situation is similar to data compression with the exception that additional side information is available at the decoder (in terms of a correlated random variable Y), that is not available at the encoder. This scenario is in fact a special case of the Slepian-Wolf problem [12]. If the side-information was also available at the encoder, clearly the encoder could take advantage of this since it could reproduce the behaviour of the decoder. It is not immediately clear how the performance changes if the side-information is only available at the decoder. The surprising result of Slepian and Wolf shows that even if the side-information is not available at the encoder, the optimal achievable performance is just as good as if it were.

Suppose now that one can tolerate some distortion in the reproduction of X (while keeping s_1 open and s_2 closed). This problem, known as the Wyner-Ziv problem [13, 14], is the rate distortion equivalent to the previously outlined side-information problem. Wyner and Ziv showed analogously that for Gaussian sources the optimal performance achievable without side-information at the encoder is the same as if the side-information was available at the encoder.

Although point-to-point source and channel coding are well known and solved problems,

Table 1.1 Summary of problems dealt with in this thesis.

Problem	S_1	S_2	Channel	Coded symbols	Distortion in reconstructed $\hat{X}$
data compression	0	0	noiseless	known	no
noise robust data compression	0	0	noisy	noisy	no
Slepian-Wolf problem	0	1	noiseless	known	no
Wyner-Ziv problem	0	1	noiseless	known	yes

the problems of single step noise robust compression, separate encoding of correlated sources (the Slepian-Wolf problem) and rate-distortion coding with side-information at the decoder are still to a large extent open.

Based on encoders with trellis structure, turbo codes are among the most powerful channel coding techniques known today and have performed close to the optimal performance theoretically achievable of any code over a wide class of channels. All the outlined cases could be solved by any powerful code that incorporates soft probabilities on the message and coded symbols. Due to their highly structured form, encoders with trellis structure (i.e. turbo codes) lend themselves naturally to the inclusion of soft information on both the message and coded symbols in the decoding algorithm. The objective of this thesis is to demonstrate that turbo codes can be suitably designed for all the problems in Table 1.1 by proposing a unified framework in which coding for them is very nearly identical. Although there are other classes of codes that exhibit excellent performance and may also incorporate soft information on the message and coded symbols, this thesis will limit its scope to the issue of designing a particular class of parallel concatenated codes based on finite fields and Latin squares. As a final note, since the codes considered in this thesis are fixed-length to fixed-length, the coding techniques proposed here are probabilistic in nature [15] in the sense that reliable communication is only possible if the source statistics are employed in the decoding process.

1.3 Original Contribution of this Thesis

Chapter 2 extensively reviews published coding techniques for the relevant problems. Much research is actively pursued on these (especially the Slepian-Wolf and Wyner-Ziv problems) and many of the existing coding techniques referenced in this thesis have appeared during

the production of this work.

The original contribution of this thesis is twofold. The first is the unified framework that allows one to encode for the various source coding problems summarized in table 1.1 with practical complexity. The second is the explicit code design of the parallel concatenated trellises. As opposed to the vast majority of the turbo codes literature, the codes designed in this thesis are non-systematic, non-linear, non-binary, non-symmetric and jointly designed. Most of the work presented in this thesis has been published or submitted for publication in [16–23]:

- Initial code design ideas, with application to the Slepian-Wolf problem, may be found in [16].
- Most of the code design of Chapter 3 was presented in [17] and the complete design is in preparation for submission in [18]. Up to date results using the finalized code design as applied to the Slepian-Wolf problem may be found in [19].
- The application of turbo codes to data compression was first presented in [20] and was submitted in [21].
- The application of turbo codes to noise robust compression may be found in [22].
- The application of turbo codes to the Wyner-Ziv problem may be found in [23].

1.4 Thesis Organization

This thesis is organized as follows. Chapter 2 extensively reviews the relevant notions of information theory, turbo codes and existing coding techniques for the relevant problems. Chapter 3 presents a complete construction of the trellises utilized in the parallel concatenated (turbo) code. Chapter 4 presents simulation results and Chapter 5 summarizes this work and discusses future directions. The trellises that were utilized in the simulations may be found in Appendix A, while some aspects of the proofs of Chapter 3 and a review of Latin squares have been placed in Appendix B. A description and an algorithmic construction of the spread random interleaver utilized in the simulations may be found in Appendix C.

Chapter 2

Preliminaries on Information Theory and Coding

Since its inception, information theory has been an invaluable tool for communication engineers to establish fundamental limits on the transmission, processing and utilization of information. Before addressing these issues, one must rigourously define the terms: information, uncertainty, reliable communication, etc.

This chapter seeks to define these terms and provide a suitable background to evaluate the performance of the systems simulated in Chapter 4 by comparing them to the Optimum Performance Theoretically Achievable (OPTA). The current chapter also briefly reviews the concepts of trellis and turbo codes upon which the code design of Chapter 3 rests. Finally, a review of relevant published work that has been done in data compression, noise robust data compression as well as Slepian-Wolf and Wyner-Ziv coding is presented.

2.1 Entropy and Typical Sequences

In his revolutionary publication "A Mathematical Theory of Communication", Claude E. Shannon proposed a new paradigm for communication [6, 7]. Prior to this work, communication engineers had little formal understanding of what a message was and a vague understanding of how to transform a message into a waveform for transmission over a channel. At the time, there was only a rudimentary comprehension of the basic modulation techniques we take for granted today such as amplitude modulation (AM), frequency modulation (FM) and pulse code modulation (PCM) [24]. There was almost no basis for

comparing them and even less for evaluating how well they could perform in principle. In short, a theory of communication was non-existent and the concept of separating what was being transmitted from the act of transmitting had yet to be formalized [25]. From a communication engineering point of view, the act of transmitting, sometimes with a radio link or recording media such a vinyl record, was achieved by modulating the analog message itself.

Shannon advocated the transmission of a discrete set of signals, separating the transmitted waveform from the meaning of the message. He proved that under such a scheme, one could transmit "information" reliably over noisy channels and in essence ushered in the digital era [25]. Intuitively, if only a finite number of possible waveforms is transmitted over some channel, there is better hope of recovering which waveform was sent since one could in principle compare the received signal against each possible transmission and choose the best. Shannon formalized this idea in his theory of information [6, 7].

At the core of information theory lies the notion of entropy and mutual information. These notions can be shown to arise naturally out of the desire to simply count how many possible "typical" sequences a random variable may produce when repeated trials are generated. The fact that not all sequences occur with meaningful probability follows from the weak law of large numbers. Consider, for example, the independent flipping of a biased coin such that the probability of heads is 0.9, i.e. $P[H] = 0.9$. One expects that if the coin is flipped often enough and a running sum of the number of occurrences of heads is kept, that roughly 90% of the flips will be found to be heads. The intuition behind this reasoning is in some sense the fundamental theorem in information theory and is known as the Asymptotic Equipartition Property (AEP).

Formally, consider a memoryless random variable[1] X over a discrete alphabet $\mathcal{X} = \{a_1, ..., a_M\}$. With each output a_i of the source, one has a probability mass $P[X = a_i]$ which will be abbreviated as p_i. Construct a series of n independent trials and denote the result of the jth trial as x_j and the results of the n trials by the vector $\mathbf{x} \in \mathcal{X}^n$.

In the review of the basic information theoretic results and concepts that follows, the expositions of [26, 27] will be followed. Let $N(a_m|\mathbf{x})$ denote the number of times that the

[1] In this work, all random processes are assumed to be stationary and ergodic.

symbol a_m appears in the sequence $\mathbf{x}$. It is interesting to evaluate the quantity

$$\frac{1}{N}\log P[\mathbf{X}=\mathbf{x}] = \sum_{m=1}^{M} \frac{N(a_m|\mathbf{x})}{n}\log(P_m). \tag{2.1}$$

However, due to the weak law of large numbers [28], one has that

$$\frac{N(a_m|\mathbf{x})}{n} \underset{n\to\infty}{\longrightarrow} p_m, \tag{2.2}$$

where convergence is in probability [27]. Applying (2.2) to (2.1) yields

$$-\frac{1}{N}\log P[\mathbf{X}=\mathbf{x}] = -\sum_{m=1}^{M} p_m \log(p_m) \tag{2.3}$$

$$\triangleq H(X), \tag{2.4}$$

where the following definition has been made:

Definition 2.1.1 *Let a discrete memoryless source X with i.i.d. outputs have an alphabet $a_1, ..., a_M$. The entropy of the source X, denoted as $H(X)$, is defined to be*

$$H(X) = -\sum_{m=1}^{M} P[X=a_m]\log P[X=a_m]. \tag{2.5}$$

The following result, known as the Asymptotic Equipartition Property (AEP) has now been proved [26].

Theorem 2.1.1 *(AEP): If X is a discrete source whose outputs are i.i.d., then*

$$P[\mathbf{X}=\mathbf{x}] \underset{N\to\infty}{\longrightarrow} 2^{-nH(X)}. \tag{2.6}$$

The significance of the above result is that it is true for *almost every realization* $\mathbf{x}$ that has been chosen randomly according to the probability mass function (PMF). Intuition suggests that with high probability, the source produces one of only $2^{nH(X)}$ different sequences instead of choosing its output from all the M^n possible sequences. Somehow, the other sequences do not occur because the empirical distribution is not "close enough" to

the expected statistical distribution. In some sense, the other sequences are "not typical". This idea can be defined rigourously as follows:

Definition 2.1.2 *Let* $\mathbf{x}$ *be a sequence of length* n *produced by a discrete source with i.i.d. distributed outputs. The sequence* $\mathbf{x}$ *is said to be* ϵ*-typical, denoted as* $\mathbf{x} \in T_n^X(\epsilon)$*, if it has the following property*

$$2^{-n(H(X)+\epsilon)} \leq P[\mathbf{X} = \mathbf{x}] \leq 2^{-n(H(X)-\epsilon)}. \tag{2.7}$$

The set of all ϵ-typical sequences is denoted by $T_n^X(\epsilon)$. The following result establishes the connection between counting the number of typical sequences produced by a source X and the entropy $H(X)$ of the source [27].

Theorem 2.1.2 *For* n *sufficiently large,*

1. *If* $\mathbf{x} \in T_n^X(\epsilon)$*, then* $H(X) - \epsilon \leq -\frac{1}{n}\log P[\mathbf{X} = \mathbf{x}] \leq H(X) + \epsilon$.
2. $P[\mathbf{X} \in T_n^X(\epsilon)] > 1 - \epsilon$ *for* n *sufficiently large.*
3. $|T_n^X(\epsilon)| \leq 2^{n(H(X)+\epsilon)}$.
4. $|T_n^X(\epsilon)| \geq (1-\epsilon)2^{n(H(X)-\epsilon)}$.

Proof: Property 1 follows from definition 2.1.2. Property 2 is a consequence of Theorem 2.1.1 from which one has that for any $\delta > 0$, there exists an n_0 such that when $n > n_0$,

$$P\left[\left|-\frac{1}{n}\log P[\mathbf{X} = \mathbf{x}] - H(X)\right| \leq \epsilon\right] > 1 - \delta(n_0). \tag{2.8}$$

The desired result is obtained by setting $\delta \leq \epsilon$ which can be done by choosing n_0 sufficiently large. Property 3 can be obtained through the following chain of equalities:

$$\begin{aligned} 1 &\geq P[\mathbf{X} \in T_n^X(\epsilon)] && (2.9)\\ &= \sum_{\mathbf{x} \in T_n^X(\epsilon)} P[\mathbf{X} = \mathbf{x}] && (2.10)\\ &\geq \sum_{\mathbf{x} \in T_n^X(\epsilon)} 2^{-n(H(X)+\epsilon)} && (2.11)\\ &= 2^{-n(H(X)+\epsilon)}|T_n^X(\epsilon)|. && (2.12) \end{aligned}$$

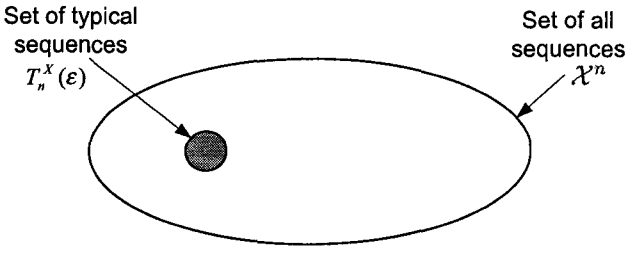

Fig. 2.1 A graphical representation of the typical set $T_n^X(\epsilon)$. There are exponentially fewer elements in $T_n^X(\epsilon)$ than in $\mathcal{X}^n$.

Property 3 can then be obtained directly from (2.12) while property 4 can be obtained in a similar fashion starting from the relation $1 - \epsilon \leq P[\mathbf{X} \in T_n^X(\epsilon)]$ for n sufficiently large. □

The importance of the above result cannot be over-emphasized. It essentially states that with high-probability all outputs are typical and the probability of each typical output is very nearly the same. Second, it gives a rather tight bound on the number of typical sequences; there is essentially an exponential number of them with the exponential growth factor in the exponent given by $H(X)$. It follows that $H(X)$ can be related intuitively to the number of typical sequences and it is not uncommon for $H(X)$ to be though of as measuring the uncertainty of X. Fig. 2.1 illustrates graphically the relationship between the typical set $T_n^X(\epsilon)$ and the set of all sequences $\mathcal{X}^n$. Here, the typical set is intentionally made small to emphasize the fact that it has exponentially fewer elements than $\mathcal{X}^n$.

The results established above generalize in the obvious way to several random variables [27, 29]. For example, the entropy of X and Y, $H(X,Y)$, and the conditional entropy of X given Y, $H(X|Y)$, are defined as follows:

Definition 2.1.3 *The joint entropy of a pair of discrete random variables (X,Y) with distribution $p_{X,Y}(x,y)$ is*

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log p_{X,Y}(x,y). \tag{2.13}$$

Definition 2.1.4 *The conditional entropy $H(X|Y)$ for a pair of discrete random variables*

(X, Y) with distribution $p_{X,Y}(x, y)$ is defined as

$$H(X|Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{X,Y}(x, y) \log p_{X|Y}(x|y). \tag{2.14}$$

One can also define joint-typicality for two random variables in the obvious way as follows:

Definition 2.1.5 *The two sequences $(\mathbf{x}, \mathbf{y})$ are jointly ϵ-typical if $\mathbf{x} \in T_n^X(\epsilon)$, $\mathbf{y} \in T_n^Y(\epsilon)$ and*

$$2^{-n(H(X,Y)+\epsilon)} \leq p_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y}) \leq 2^{-n(H(X,Y)-\epsilon)}. \tag{2.15}$$

Theorem 2.1.2 can be extended to joint entropies $H(X, Y)$ in the obvious fashion by considering the pair of random variables (X, Y) as a single random variable Z. A bound on the conditional probability $p_{\mathbf{Y}|\mathbf{X}}$ can be obtained by dividing term by term Eq. (2.15) and Eq. (2.7) to obtain:

$$2^{-n(H(Y|X)+\epsilon)} \leq p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) \leq 2^{-n(H(Y|X)-\epsilon)}. \tag{2.16}$$

This subsection is concluded with a brief discussion on differential entropy, the extension of entropy to continuous sources. Some authors, such as Gallager [29], prefer not to define differential entropy and directly extend the definition of mutual information to continuous random variables.

Definition 2.1.6 *A continuous random variable X with probability density function $f(x)$ and a support set S is said to have a differential entropy $h(X)$ given by*

$$h(X) = -\int_S f(x) \log f(x) dx. \tag{2.17}$$

A very important result in information theory is that among all continuous distributions given a fixed finite variance σ^2, the normal distribution $X \sim N(m, \sigma^2)$ has the largest differential entropy and can be shown to be $h(X) = \frac{1}{2} \log 2\pi e\sigma^2$ [27].

All the results obtained for the entropy of discrete sources can be generalized to continuous sources. Notably, the AEP still holds and can be expressed as $-\frac{1}{n} \log f(\mathbf{X}) \to h(X)$ and a typical set can be suitably defined as $T_n^X(\epsilon) = \{\mathbf{x} \in S^n : |-\frac{1}{n} \log f(\mathbf{x}) - h(X)| \leq \epsilon\}$ for which $Pr[\mathbf{X} \in T_n^X(\epsilon)] > 1 - \epsilon$, analogous to Theorem 2.1.2, also holds [27]:

2.2 Mutual Information and Channel Capacity in AWGN

Consider a discrete memoryless source generating a large number of outputs n, with transmission over a discrete-time memoryless channel modelled by the transition probabilities $p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \prod_i p_{Y|X}(y_i|x_i)$, as illustrated in Fig. 2.2.

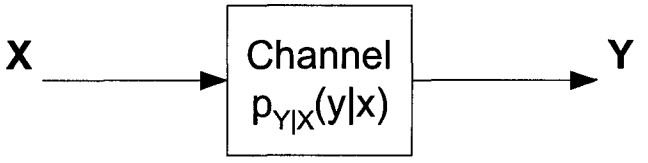

Fig. 2.2 Transmission of information over a channel.

Suppose one were to model the uncertainty associated with $\mathbf{X}$ by the number of bits needed to count all the typical sequences [26]

$$\log |T_n^X(\epsilon)| \simeq \log 2^{nH(X)} = nH(X), \tag{2.18}$$

while the uncertainty left after observing $\mathbf{Y} = \mathbf{y}$ can be expressed as

$$\log |T_n^{X|Y}(\epsilon)| \simeq \log 2^{nH(X|Y)} = nH(X|Y). \tag{2.19}$$

The reduction of the uncertainty associated with $\mathbf{X}$ at the channel output is then

$$\log \frac{|T_n^X(\epsilon)|}{|T_n^{X|Y}(\epsilon)|} \simeq nH(X) - nH(X|Y) \tag{2.20}$$

$$\triangleq nI(X;Y). \tag{2.21}$$

The term $I(X;Y)$ is called the mutual information between X and Y and it is easy to verify that $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$. Analogously, the mutual information $I(X;Y)$ of two continuous sources is defined as $I(X;Y) = h(X) - h(X|Y) = h(Y) - h(Y|X)$.

Intuitively, if the uncertainty on vector $\mathbf{X} = (X_1, ..., X_n)$ were reduced by disallowing sufficiently many n-length sequences from the source, as illustrated in Fig. 2.3, the uncertainty $H(\mathbf{X})$ would be *below* the reduction $I(\mathbf{X};\mathbf{Y}) = nI(X,Y)$ otherwise possible at the decoder. This concept, often called sphere packing, suggests the possibility of reliably

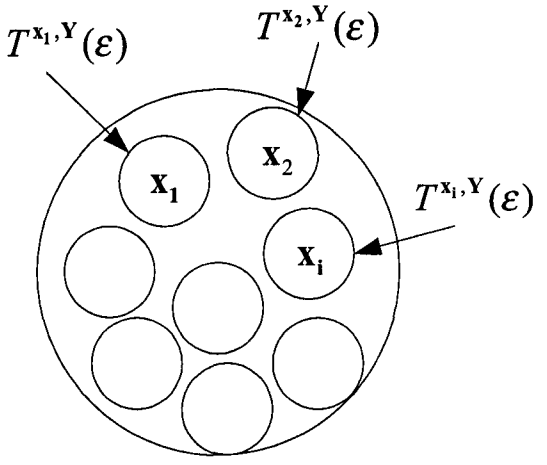

Fig. 2.3 Sphere packing for discrete memoryless channels.

communicating one of a possible set of sequences over the unreliable "probabilistic" channel and motivates the following definition [27].

Definition 2.2.1 *The information capacity of the discrete-time memoryless channel is defined as*

$$C = \max_{p(x)} I(X;Y), \tag{2.22}$$

where the maximization is performed over all distributions on X.

Example:

Consider, as illustrated in Fig. 2.4, the information capacity of the Gaussian channel where $Y = X + Z$ with $Z \sim N(m, \sigma_N^2)$ and a power constraint $E[X^2] \leq P$ is imposed. One then has,

$$\begin{aligned} I(X;Y) &= h(Y) - h(Y|X) && (2.23)\\ &= h(Y) - h(X+Z|X) && (2.24)\\ &= h(Y) - h(Z|X) && (2.25)\\ &= h(Y) - h(Z) && (2.26)\\ &= h(Y) - \frac{1}{2}\log 2\pi e\sigma_N^2 && (2.27)\\ &\leq \frac{1}{2}\log 2\pi e(P+\sigma_N^2) - \frac{1}{2}\log 2\pi e\sigma_N^2. && (2.28) \end{aligned}$$

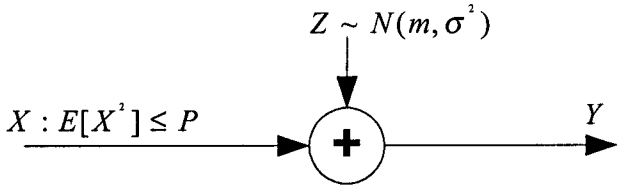

Fig. 2.4 The additive white Gaussian noise channel.

The inequality follows from the fact that $VAR[Y^2] = P+\sigma_N^2$ and the differential entropy of a continuous random variable Y is at most $\frac{1}{2}\log 2\pi e\sigma_y^2$ with equality when Y is Gaussian. Therefore, one has that,

$$C = \max_{p(x):E[X^2]\leq P} I(X;Y) = \frac{1}{2}\log(1 + P/\sigma_N^2), \tag{2.29}$$

where the maximizing distribution is $X \sim N(0, P)$.

Definition 2.2.2 *Consider the Gaussian channel with power constraint P. An (M, n) code is defined by an index set $\mathcal{M} = \{1, ..., M\}$, an encoding function $f : \mathcal{M} \rightarrow \mathcal{X}^n$ such that $\sum_{i=1}^n f_i^2(j) \leq nP$ and a decoding function $g : \mathcal{Y}^n \rightarrow \mathcal{M}$.*

Definition 2.2.3 *A rate R is achievable if there exists a sequence of $(2^{nR}, n)$ codes and the maximal probability of error, $\lambda_n = \max_{j\in\mathcal{M}} P[g(f(j) + \mathbf{Z}) \neq j]$, tends to zero. The channel capacity is defined as the supremum of all achievable rates.*

Theorem 2.2.1 *The channel capacity of the Gaussian channel is the information capacity $C = \frac{1}{2}\log(1 + P/\sigma_N^2)$.*

Only the proof that the rate is achievable is presented here as it provides insight into the code design of Chapter 3 with its random coding argument. The converse to the channel coding theorem for Gaussian channels: that a rate above the information capacity yields a strictly positive probability of error, generally requires the use of Fano's inequality and can be found in [27, 29].

Proof:(achievability)

- Codebook: Generate a codebook randomly according to a normal distribution. If the variance of each letter in the codebook is $P-\epsilon$, with high probability each codeword in

the codebook will satisfy the power constraint for sufficiently large n. More formally, $f_i(j) \sim N(0, P - \epsilon)$.

- Encoding: Once the codebook has been generated, it is revealed to both the encoder and the decoder. Message j is transmitted with the codeword $f(j) = (f_1(j), ..., f_n(j))$.
- Decoding: The decoder searches for a codeword $(f_1(j), ..., f_n(j))$ in the codebook that is jointly typical with the received vector $\mathbf{Y}$. If there is only one such vector, the receiver declares it as the transmitted message, otherwise an error is declared.
- Probability of error: By the symmetry of the problem, assume that the first codeword was sent: $\mathbf{Y} = f(1) + \mathbf{Z}$.

One may then define the following events: $E_0 = \{\frac{1}{n}\sum_{i=1}^{n} f_i(1) > P\}$ and $E_i = \{(f(i), \mathbf{Y}) \in T_n^{X,Y}(\epsilon))\}$. The event E_0 occurs when the codeword to be transmitted violates the power constraint. In this case, the encoder may transmit a fictitious codeword that satisfies the power constraint and an error will be made at the decoder. By the union bound on the probability of events, the probability of error can be bounded as

$$P_e^n \leq P[E_0] + P[E_1^c] + \sum_{i=2}^{2^{nR}} P[E_i], \tag{2.30}$$

where both $P[E_0] \to 0$ and $P[E_1^c] \to 0$ as $n \to \infty$ by the law of large numbers and the AEP. Now, the probability that $\mathbf{Y}$ is jointly typical with $f(j)$ when $j \neq 1$ is at most $2^{-n(I(X;Y)-\epsilon)}$ [27] and hence

$$P_e^n \leq \epsilon + \epsilon + \sum_{i=2}^{2^{nR}} 2^{-n(I(X;Y)-\epsilon)} \tag{2.31}$$

$$\leq 2\epsilon + 2^{nR} 2^{-n(I(X;Y)-\epsilon)} \tag{2.32}$$

$$\leq 2\epsilon + 2^{\epsilon} 2^{-n(I(X;Y)-R)}, \tag{2.33}$$

which goes to zero if $R < I(X;Y) - \epsilon$. It still needs to be shown that the maximal probability of error goes to zero. Unfortunately, this may not always be the case. However, no more than half the codewords may have a probability of error above twice the average. Since the average goes to zero, at least half the codewords have a probability of error

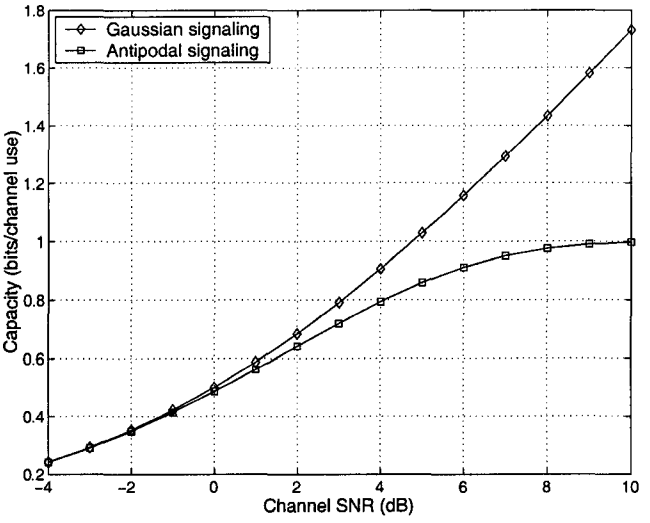

Fig. 2.5 A comparison of AWGN channel capacity for Gaussian and Antipodal signaling.

that also goes to zero. If we limit ourselves to utilizing these codewords, then we have a $(2^{nR-1}, n)$ codebook for which the rate can be made arbitrarily close to the capacity C for sufficiently large n.

□

Suppose that the transmitter cannot produce a continuous range of amplitudes for transmission over the AWGN channel, but is instead restricted to sending one of two signal levels from the set $\{-1, 1\}$. This antipodal signaling is often referred to as Binary Phase Shift Keying (BPSK) since it models exactly the output of a demodulator receiving one of two sine waves phase shifted 180 degrees apart. It is equally interesting to investigate the capacity of such a system and to compare against the previous result. The capacity for antipodal signaling is again $C = \max_{p(x)} I(X;Y)$ where the maximizing distribution is the uniform distribution. Fig. 2.5 compares capacity functions for both signaling methods.

For low channel SNRs, there is little loss in terms of information bits transmitted per channel use. At high SNRs, there is quite a performance gap. Intuitively, it is clear that antipodal signaling cannot transmit more than 1 bit/channel use no matter how high the quality of the channel. Gaussian signaling on the other hand is not limited in this way. As a figure of comparison, to transmit 0.8 bits/channel use, antipodal signaling requires roughly 1.0dB more in channel SNR compared to Gaussian signaling. As a final note, if the

channel SNR is sufficiently high, higher order modulation techniques are often employed to achieve transmission rates above 1 bit/channel use.

2.3 Random Binning and Data Compression

The concept of data compression is well known and can be found to be implemented in practice in popular programs such as gzip, WinRar and LZW. The possibility to do data compression can be explained by information theory.

Consider the set off all M-ary sequences of length n. By the pigeonhole principle, it is clearly impossible to uniquely represent all of these with anything less than $n \log M$ bits. However, from the information theory perspective, not all of these sequences occur with meaningful probability. Of the $2^{n \log M}$ possible sequences, only $2^{nH(X)}$ are likely to occur and this suggests that about $nH(X)$ bits should be enough to represent all probable outputs of the source. The remaining sequences occur arbitrarily rarely as the packet size n is increased.

A $(2^{nR}, n)$ source code is specified by an encoder $i_X = f(\mathbf{x})$ which maps $\mathcal{X}^n$ to the set of integers $\mathcal{M} = \{1, ..., 2^{nR}\}$, and a decoder $g : \mathcal{M} \rightarrow \mathcal{X}^n$.

Definition 2.3.1 *For a given source, a rate R is said to be achievable if there exists a sequence of $(2^{nR}, n)$ source codes with $P_e = P[g(f(\mathbf{X})) \neq \mathbf{X}] \rightarrow 0$ as $n \rightarrow \infty$. The achievable rate region is the closure of the set of achievable rates.*

Theorem 2.3.1 *The achievable rate region for a source X is $R \geq H(X)$.*

Proof:(achievability) Although the intuitive argument above can be formalized, a proof of data compression using the random binning argument [27] is provided instead, as it motivates the code design rules in Chapter 3. Instead of uniquely assigning a codeword to each typical sequence of the source, one randomly assigns codewords to each sequence of the source. More formally, let $f(\mathbf{X})$ be a mapping selected randomly and *uniformly* so that $f : \mathcal{X}^n \rightarrow \mathcal{M} = \{1, ..., 2^{nR}\}$. Given $f(.)$, the decoder $g : \mathcal{M} \rightarrow \mathcal{X}^n$ chooses a typical sequence in $\mathcal{X}^n$ which maps to the correct bin. It declares and error if either no such sequence can be found or if there is more than one such sequence. The probability of error, averaged over the random choice of the mapping f, can be bounded as [27]:

$$P[g(f(\mathbf{X})) \neq \mathbf{X}] \leq P[\mathbf{X} \notin T_n^X(\epsilon)] + P[\exists \mathbf{x}' \neq \mathbf{X} : f(\mathbf{x}') = f(\mathbf{X}), \mathbf{x}' \in T_n^X(\epsilon)] \qquad (2.34)$$

$$\leq \epsilon + \sum_{\mathbf{x}} P[\exists \mathbf{x}' \neq \mathbf{x} : \mathbf{x}' \in T_n^X(\epsilon), f(\mathbf{x}') = f(\mathbf{x})]p(\mathbf{x}) \tag{2.35}$$

$$\leq \epsilon + \sum_{\mathbf{x}} \sum_{\substack{\mathbf{x}' \in T_n^X(\epsilon) \\ \mathbf{x}' \neq \mathbf{x}}} P[f(\mathbf{x}') = f(\mathbf{x})]p(\mathbf{x}) \tag{2.36}$$

$$\leq \epsilon + \sum_{\mathbf{x}} \sum_{\mathbf{x}' \in T_n^X(\epsilon)} 2^{-nR} p(\mathbf{x}) \tag{2.37}$$

$$\leq \epsilon + 2^{n(H(X)+\epsilon)-nR}. \tag{2.38}$$

Provided that $R > H(X)$, then the probability of error can be made arbitrarily small. The entire region in Theorem 2.3.1 is obtained by the closore of this set.

□

The converse, that the probability of error is strictly greater than 0 when $R < H(X)$ can be shown with Fano's inequality and may be found in [27, 29].

A few important points can be made about the above. First, as opposed to the original outlined intuition, the encoder does not need to know which sequences are the typical ones, only the decoder requires this information. It is precisely this property that allows for distributed source coding. Second, one may ask why is it that the encoder does not need to know which are the typical sequences? Is not a random binning likely to result in multiple typical sequences assigned to the same bin? To answer these questions, the ratio of the number of typical sequences to the number of bins is in fact $2^{nH(X)-nR}$ and if $R > H(X)$, then there are exponentially more bins than typical sequences as n increases. In fact, it is quite unlikely that any bin has even one typical sequence.

2.4 Joint Source-Channel Coding

Consider the situation illustrated in Fig. 2.6a. There, a source X is first source coded to remove any redundancy. This is followed by a channel encoder that introduces a controlled amount of redundancy to protect the data against errors during transmission over the channel. If the rate of the source code is $R_s > H(X)$ and that of the channel code $R_c < C$, then the overall rate is $R = R_s/R_c$ channel uses/source symbol with the minimum rate of $R > H(X)/C$ required for reliable communication in the concatenated system.

A competing scenario is considered in Fig. 2.6b where a single encoder both source codes and channel codes the information source. Clearly, if the best method is to use a

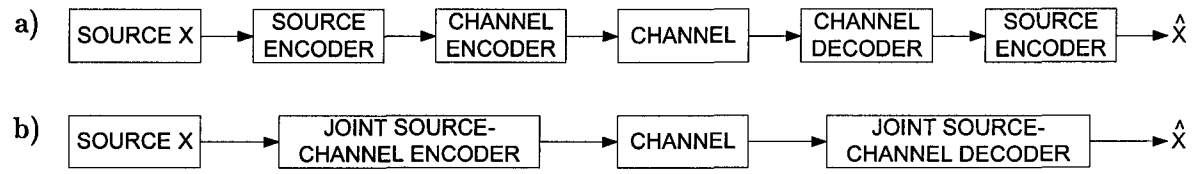

Fig. 2.6 A comparison between (a) serial concatenation of source and channel codes with (b) joint source-channel coding.

concatenation of separate source/channel encoders, the second scenario includes this as a special case so it may do no worse than the first. An interesting question is if anything can be gained by designing a joint source-channel code. The surprising answer, known as the source-channel coding theorem, says no [27, 29]. In principle, there is nothing to be gained or lost with either communication system in terms of the optimum performance that can be theoretically achieved. In practice, the complexity of a joint source-channel code is usually prohibitive compared to separate source and channel coding.

An example now follows to evaluate the capacity of a channel in terms of the ratio of the energy per source bit E_b to the noise density spectrum N_0 of an AWGN channel.

Example:

Consider the tandem scheme in Fig. 2.6. A sequence of n source symbols is encoded losslessly into a sequence of $n(H(X)+\epsilon)$ bits by a source code. The channel code can then be seen as expanding the $n(H(X)+\epsilon)$ bits into an $(2^{n(H(X)+\epsilon)}, nR)$ code, effectively operating at a rate of $R_c = (H(X)+\epsilon)/R$.

Reliable communication is possible if $H(X)/R < \frac{1}{2}\log(1+P/\sigma_N^2)$. One also has that the energy per source symbol is $E_b = RP$ while $\sigma_N^2 = N_0/2$. Together, these yield

$$\frac{H(X)}{R} < \frac{1}{2}\log\left(1+\frac{2}{R}\frac{E_b}{N_0}\right) \tag{2.39}$$

$$\frac{E_b}{N_0} > \frac{R}{2}\left(2^{\frac{2H(X)}{R}}-1\right). \tag{2.40}$$

If X is a binary i.i.d. source biased so that $P[X=0] = 0.1$ and $R = 2/3$ (2 channel uses for 3 binary source symbols), reliable communication can be achieved with Gaussian signaling provided $E_b/N_0 > -2.59$ dB.

The required E_b/N_0 for binary signaling is now obtained by determining the gap in performance between Gaussian and antipodal signaling. In the tandem scheme, the channel

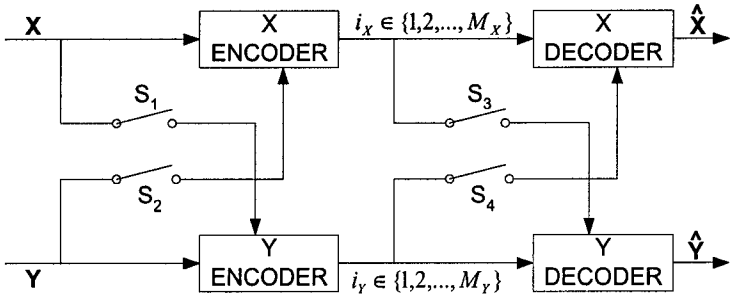

Fig. 2.7 Schematic block diagram of the 16 cases that Slepian and Wolf considered.

code transmits $H(X)/R = 0.7035$ bits/channel use. From Fig. 2.5, binary signaling exhibits a gap of 0.61 dB against Gaussian signaling and a minimum of $E_b/N_0 > -1.98$ dB is needed.

2.5 The Slepian-Wolf Problem

Consider two possibly correlated sources of information, (X, Y), each generating a sequence of outputs ..., X_{-1}, X_0, X_1, ... and ..., Y_{-1}, Y_0, Y_1, ... If each source is taken separately and is encoded/decoded without knowledge of the other, clearly the minimum rate needed to encode these two sources is $H(X) + H(Y)$. In the case that X and Y are not independent, joint source coding/decoding can reduce the rate to $H(X,Y) \leq H(X) + H(Y)$.

Of particular interest are the "in-between" cases where there is neither complete joint nor complete separate encoding/decoding of the sources. This was first investigated by Slepian and Wolf for the 16 cases that could be described by the setting of switches S_1 through S_4 in Fig. 2.7 [12]. By far the most interesting setting is 0011, illustrated in Fig. 2.8a, which has now become synonymous as the Slepian-Wolf problem.

It is not immediately clear what pair of rates (R_X, R_Y) are required for reliable communication, though it was shown that the region of Fig. 2.8b is in fact achievable. Before proceeding to prove this, a few definitions are required.

Let X and Y be discrete memoryless sources whose outputs are from the sets $\mathcal{X} = \{a_1, ..., a_{M_X}\}$ and $\mathcal{Y} = \{a_1, ..., a_{M_Y}\}$ respectively. An X-encoder $i_X = f_X(\mathbf{x})$ is a single-valued function from the set $\mathcal{X}^n$ to the set of integers $\mathcal{M}_X = \{1, 2, ..., 2^{nR_X}\}$. Likewise, a Y-encoder $i_Y = f_Y(\mathbf{y})$ is a single valued function from the set $\mathcal{Y}^n$ to the set of integers $\mathcal{M}_Y = \{1, 2, ..., 2^{nR_Y}\}$. A decoder g is a mapping $g : \mathcal{M}_X \times \mathcal{M}_Y \rightarrow \mathcal{X}^n \times \mathcal{Y}^n$.

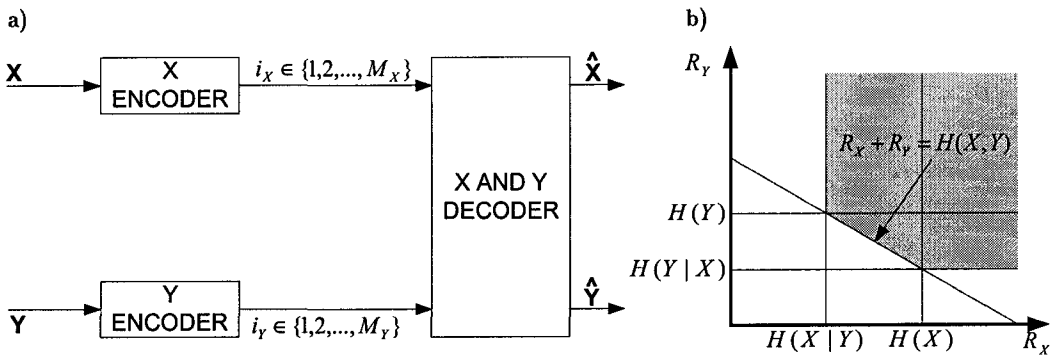

Fig. 2.8 The (a) schematic block diagram of the Slepian-Wolf problem and (b) the associated achievable region.

The pair of encoders f_X and f_Y are said to form a $((2^{nR_X}, 2^{nR_Y}), n)$ *distributed source code.* The probability of error for the distributed source code is then

$$P_e = P[g(f_X(\mathbf{X}), f_Y(\mathbf{Y})) \neq (\mathbf{X}, \mathbf{Y})] \tag{2.41}$$

Definition 2.5.1 *A rate pair (R_X, R_Y) is said to be achievable if there exists a sequence of $(2^{nR_X}, 2^{nR_Y})$ codes with $P_e \to 0$ as $n \to \infty$. The achievable rate region is the closure of the set of achievable rates.*

Theorem 2.5.1 *The achievable rate region for the problem illustrated in Fig. 2.8a is given by $R_X \geq H(X|Y)$, $R_Y \geq H(Y|X)$ and $R_X + R_Y \geq H(X,Y)$.*

Proof:[2] Each encoder assigns an integer from $\{1, ..., 2^{nR_X}\}$ and $\{1, ..., 2^{nR_Y}\}$, randomly and uniformly to every sequence in $\mathcal{X}^n$ and $\mathcal{Y}^n$. If there is one and only one jointly typical sequence $(\mathbf{x}, \mathbf{y})$ that is mapped to the pair of integers specified by the encoders, then that sequence is our estimate at the decoder, otherwise an error is declared. There are four error events:

$$E_0 = \{(\mathbf{X}, \mathbf{Y}) \notin T_n^{X,Y}(\epsilon)\},$$
$$E_1 = \{\exists \mathbf{x}' \neq \mathbf{X} : f_X(\mathbf{x}') = f_X(\mathbf{X}), (\mathbf{x}', \mathbf{Y}) \in T_n^{X,Y}(\epsilon)\},$$
$$E_2 = \{\exists \mathbf{y}' \neq \mathbf{Y} : f_Y(\mathbf{y}') = f_Y(\mathbf{Y}), (\mathbf{X}, \mathbf{y}') \in T_n^{X,Y}(\epsilon)\},$$
$$E_3 = \{\exists (\mathbf{x}', \mathbf{y}') : \mathbf{x}' \neq \mathbf{X}, \mathbf{y}' \neq \mathbf{Y}, f_X(\mathbf{x}') = f_X(\mathbf{X}), f_Y(\mathbf{y}') = f_Y(\mathbf{Y}), (\mathbf{x}', \mathbf{y}') \in T_n^{X,Y}(\epsilon)\}.$$

[2]The original proof in [12] requires analytical methods. The first proof based on random binning was published in [30] when generalizing to ergodic sources and is summarized here.

The first error event, E_0, corresponds to the case where the sequences generated by the sources are not jointly typical. The second error event, E_1, occurs when there is a sequence $\mathbf{x}$ that is different from the one that was actually generated (i.e. $\mathbf{X}$) and is indistinguishable from $\mathbf{X}$ as far as the decoder is concerned. In other words, $\mathbf{x}$ has the same encoding as $\mathbf{X}$ and is jointly typical with $\mathbf{Y}$. The error event E_2 is similar to that of E_1. Finally, E_3 occurs when there is a pair $(\mathbf{x}, \mathbf{y})$ with both sequences different from the actual sequences $(\mathbf{X}, \mathbf{Y})$ that were generated and which again cannot be distinguished at the decoder. In this last case, both the encodings for $\mathbf{x}$ and $\mathbf{y}$ are the same as those for $\mathbf{X}$ and $\mathbf{Y}$ as well as $(\mathbf{x}, \mathbf{y})$ is a typical pair.

By the union bound of events, $P_e \leq P[E_0] + P[E_1] + P[E_2] + P[E_3]$. That $P[E_0] \to 0$ as $n \to \infty$ follows by the AEP. $P[E_1]$ and $P[E_2]$ are similar with

$$P[E_1] \leq \sum_{(\mathbf{x},\mathbf{y})} p(\mathbf{x},\mathbf{y}) \sum_{\substack{\mathbf{x}' \neq \mathbf{x} \\ (\mathbf{x}',\mathbf{y}) \in T_n^{X,Y}(\epsilon)}} P[f_X(\mathbf{x}') = f_X(\mathbf{x})] \tag{2.42}$$

$$\leq \sum_{(\mathbf{x},\mathbf{y})} p(\mathbf{x},\mathbf{y}) 2^{-nR_X} |T_n^{X|Y}(\epsilon)| \tag{2.43}$$

$$\leq 2^{-n(R_X - H(X|Y) - \epsilon)}. \tag{2.44}$$

The above bound goes to zero when $R_X > H(X|Y)$ and n is taken sufficiently large. Finally, $P[E_3]$ may be bounded as

$$P[E_3] \leq \sum_{(\mathbf{x},\mathbf{y})} p(\mathbf{x},\mathbf{y}) \sum_{\substack{\mathbf{x}' \neq \mathbf{x}, \mathbf{y}' \neq \mathbf{y} \\ (\mathbf{x}',\mathbf{y}') \in T_n^{X,Y}(\epsilon)}} P[f_X(\mathbf{x}') = f_X(\mathbf{x}), f_Y(\mathbf{y}') = f_Y(\mathbf{y})] \tag{2.45}$$

$$\leq \sum_{(\mathbf{x},\mathbf{y})} p(\mathbf{x},\mathbf{y}) 2^{-n(R_X + R_Y)} |T_n^{X,Y}(\epsilon)| \tag{2.46}$$

$$\leq 2^{-n(R_X + R_Y - H(X,Y) - \epsilon)}. \tag{2.47}$$

It remains to show the converse, i.e. that the points outside the established achievable region are not themselves achievable. It is quite clear that $R_X + R_Y < H(X,Y)$ is not achievable since this would imply the ability to code a single source $Z = (X,Y)$, below the entropy $H(Z) = H(X,Y)$. Now suppose that there exists a rate R_X such that with $R_Y = H(Y|X) - \delta$, an achievable pair is formed. Then, one can replace the encoder f_X with another encoder f_X' such that $R_X = H(X) + \delta/2$ since then the source X may be

encoded error free. Hence, $(H(X) + \delta/2, H(Y|X) - \delta)$ is also achievable, but $R'_X + R_Y = H(X,Y) - \delta/2 < H(X,Y)$ and a contradiction is reached.

□

Now that it has been established that for the region of Fig. 2.8b, a probability of error of 0 can be asymptotically reached, the next logical question is for what region can an error of exactly zero be realized. As shown by Orlitsky [31], the answer is not characterized by any of the various entropies of the sources but requires knowledge of the joint distribution $p(x,y)$.

Theorem 2.5.2 *If $p(x,y) > 0$ and an error probability of exactly 0 is desired, the achievable rate points in the Slepian-Wolf problem satisfy $R_X \geq H(X)$ and $R_Y \geq H(Y)$.*

Further results on the Slepian-Wolf problem can be found in the works of Csiszár [32] and Oohama et al [33]. Csiszár investigated exponential bounds on the probability of error as a function of n using the method of types [34] while Oohama et al. considered the issue of universal coding where neither the encoder nor the decoder has explicit knowledge of the statistics of the source.

2.6 Rate Distortion Theory and the Wyner-Ziv Problem

A natural extension of reliable communication is the transmission of information subject to some fidelity criterion. For example, the transmission of a real number typically requires an infinite number of bits. If a fidelity criterion that does not require perfect reconstruction of the number at the receiver is assumed, then it may yet be possible to achieve this with a finite number of bits.

Rate distortion theory involves the problem of quantization. Consider a scalar random variable $X \sim N(0,1)$ representing a voltage. It is desired to represent this voltage as accurately as possible subject to the criterion that only 1 of two levels is actually stored. If these levels are called $\hat{\mathcal{X}} = \{a_1, a_2\}$, then the quantizer is represented by the mapping $Q : \mathcal{R} \rightarrow \hat{\mathcal{X}}$. As is, there is no reason to prefer one pair of quantization levels over another. However, suppose one seeks to minimize the mean squared error $E[d(X, Q(X)] = E[X - Q(X)]^2$. The choice of quantization levels is no longer arbitrary and depends on the distribution of X.

Motivated by the above example, one has the following definitions.

Definition 2.6.1 *A distortion measure is a mapping* $d : \mathcal{X} \times \hat{\mathcal{X}} \rightarrow [0, \infty)$.

Intuitively, the distortion measure is the cost of representing x by $\hat{x}$. Examples of common distortion measures are the *Hamming distortion*, sometimes referred to as the *probability of error* measure. There,

$$d(x, \hat{x}) = \begin{cases} 0 & x = \hat{x}, \\ 1 & x \neq \hat{x}. \end{cases} \tag{2.48}$$

Alternatively, another common distortion measure is the *squared error* distortion

$$d(x, \hat{x}) = (x - \hat{x})^2. \tag{2.49}$$

Often, it is convenient to measure the distortion between the sequence $\mathbf{x}$ and its approximation $\hat{\mathbf{x}}$. This is usually done with the additive distortion measure $d(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{n} \sum_{m=1}^{n} d(x_m, \hat{x_m})$. The following definitions are from [27]:

Definition 2.6.2 *A* $(2^{nR}, n)$ *rate distortion code is an encoding* $f_n : \mathcal{X}^n \rightarrow \{1, 2, ..., 2^{nR}\}$ *and decoding* $g_n : \{1, 2, ..., 2^{nR}\} \rightarrow \hat{\mathcal{X}}^n$ *with associated distortion* $D = E[d(\mathbf{X}, g_n(f_n(\mathbf{X})))]$.

Definition 2.6.3 *A rate distortion pair* (R, D) *is said to be achievable if there exists a sequence of* $(2^{nR}, n)$ *rate distortion codes with* $\lim_{n \rightarrow \infty} E[d(\mathbf{X}, g_n(f_n(\mathbf{X})))] \leq D$.

It is interesting to characterize the locus of all achievable rate distortion pairs (R, D). This if often done with the *rate distortion function* $R(D)$, defined as follows [27]:

Definition 2.6.4 *The rate distortion region is the closure of the set of achievable pairs* (R, D).

Definition 2.6.5 *The rate distortion function* $R(D)$ *is the infimum of rates* R *such that* (R, D) *is in the rate distortion region for a given* D.

Theorem 2.6.1 *The rate distortion function is given as*

$$R(D) = \min_{\substack{P_{\hat{X}|X}(\hat{x}|x) \\ E[d(x,\hat{x})] < D}} I(X; \hat{X}). \tag{2.50}$$

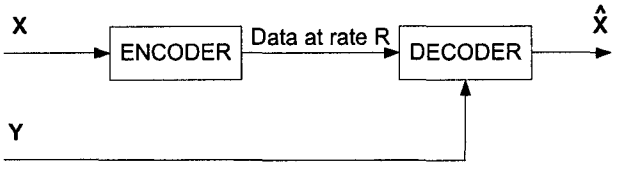

Fig. 2.9 The Wyner-Ziv problem of rate-distortion with side information at the decoder.

Example:

Consider a binary source with i.i.d. outputs. If an application can tolerate a maximum distortion D as specified by the bit error rate (BER), one may ask what minimum rate is required. In this case, the Hamming distortion measure is appropriate as it is equivalent to a BER measure. Shannon originally considered this problem [35] and the rate-distortion curve is given by

$$R(D) = \begin{cases} H(p) - H(D), & 0 \leq D \leq \min\{p, 1-p\}, \\ 0, & D > \min\{p, 1-p\}. \end{cases} \tag{2.51}$$

Example:

Wyner and Ziv originally considered the problem of rate-distortion coding with side-information at the decoder [13, 14]. The particular situation relevant here is illustrated in Fig. 2.9. In some sense, this problem can almost be considered as the rate-distortion equivalent to the Slepian-Wolf problem. If the source X is discrete, perfect reconstruction is equivalent to the Slepian-Wolf problem. Suppose that $X \sim N(m_X, \sigma_X^2)$ and $Y = X + U$ where $U \sim N(m_U, \sigma_U^2)$ and independent of X. With a squared error distortion metric, Wyner and Ziv showed that

$$R(D) = \begin{cases} \frac{1}{2}\log\frac{\sigma_X^2\sigma_U^2}{(\sigma_X^2+\sigma_U^2)D}, & 0 < D \leq \frac{\sigma_X^2\sigma_U^2}{(\sigma_X^2+\sigma_U^2)}, \\ 0, & D \geq \frac{\sigma_X^2\sigma_U^2}{(\sigma_X^2+\sigma_U^2)}. \end{cases} \tag{2.52}$$

Fig. 2.10 shows the normalized distortion, as measured by $10\log_{10}(D/\sigma_X^2)$ versus the *correlation SNR* measured in dB as $10\log_{10}\frac{\sigma_X^2}{\sigma_U^2}$ with and without side-information at the decoder. From the figure, it is clear that the side-information provides significant gains in terms of achievable distortion over a large range of correlation.

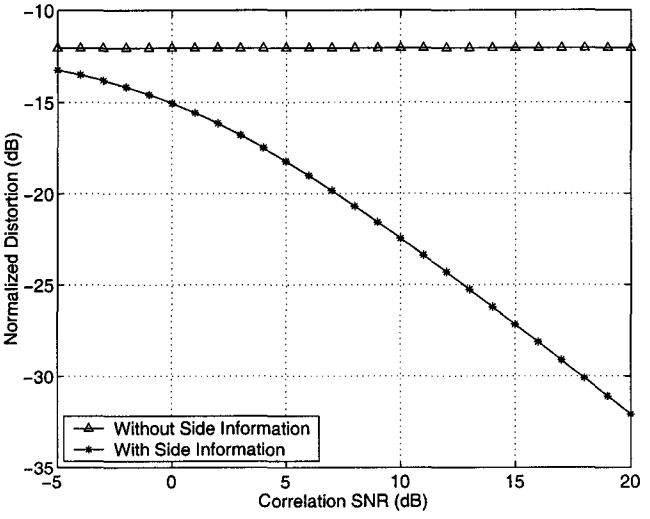

Fig. 2.10 Distortion curve for rate $R = 2$ as the correlation SNR is varied.

2.7 Encoders with Trellis Structure and the BCJR Algorithm

Trellis codes have been of interest for several decades due to their relatively good performance compared with existing alternatives, elegant structure and existence of efficient decoding algorithms. As a case in point, consider space systems where communication must be done over truly vast distances. The Mariner Mars and Viking missions, utilizing a rate 6/32 biorthogonal (Reed-Muller) code [36] required $E_b/N_0 = 6.4$ dB while Galileo utilizing a (4,1,14) convolutional code with maximum likelihood decoding necessitated only $E_b/N_0 = 1.75$ dB for the same BER of 10^{-5} (2.5 dB from capacity) [37].

Convolutional codes (CC) [1, 38] are perhaps the most pervasive form of trellis based codes. A typical *recursive* CC with two memory elements is illustrated in Fig. 2.11a. A finite memory shift register is gradually loaded with an input that depends not only on the input bit, but linearly with the state of every memory element. The output is again a linear combination of the input and memory. One may immediately observe that binary convolutional encoders are linear under modulo 2 addition. In particular, if inputs $\mathbf{X}_1$ and $\mathbf{X}_2$ generate outputs $\mathbf{U}_1$ and $\mathbf{U}_2$ respectively, then input $\alpha_1\mathbf{X}_1 + \alpha_2\mathbf{X}_2$ generates output $\alpha_1\mathbf{U}_1 + \alpha_2\mathbf{U}_2$. What makes this code attractive is that it can be completely characterized by a state transition diagram "unwrapped in time" (Fig. 2.11b) which is often called a *trellis*.

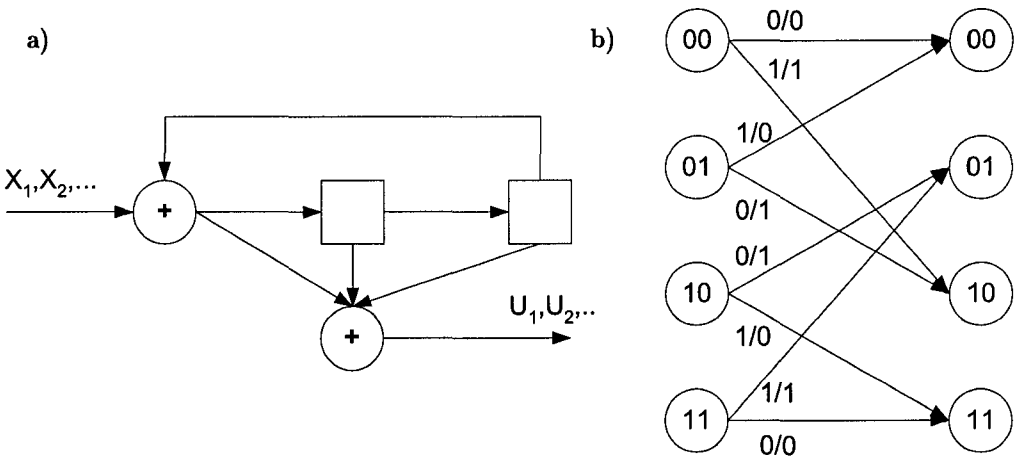

Fig. 2.11 A recursive convolutional code (a) and one stage of its associated trellis (b) with input/output edge labels denoted by X_i/U_i.

Definition 2.7.1 *A trellis section is a five-tuple* $X = (G, S, G', S', B)$, *where* G *and* G' *are the input and output alphabets respectively,* S *and* S' *are the left and right states respectively, and the branches* B *are a subset of* $S \times S' \times G \times G'$.

In Chapter 3, it will be necessary to manipulate trellises directly and a more general framework than the shift register structure is required to represent a (possibly non-linear) trellis. These trellis based encoders are called Finite State Machine (FSM) encoders. In the framework employed there an FSM encoder is described by two matrices: an input state transition matrix whose (i,j)th entry corresponds to the input sequence when the encoder makes a transition from state i to state j, and an output state transition matrix whose (i,j)th entry corresponds to the output sequence for the described transition. For example, the convolutional encoder in Fig. 2.11a is specified by the following pair of matrices

$$M_{in} = \begin{bmatrix} 0 & \phi & 1 & \phi \\ 1 & \phi & 0 & \phi \\ \phi & 0 & \phi & 1 \\ \phi & 1 & \phi & 0 \end{bmatrix}, \qquad M_{out} = \begin{bmatrix} 0 & \phi & 1 & \phi \\ 0 & \phi & 1 & \phi \\ \phi & 1 & \phi & 0 \\ \phi & 1 & \phi & 0 \end{bmatrix}, \tag{2.53}$$

where ϕ represents a dummy symbol that is never input into the FSM.

Due to their graph structure, there are numerous efficient algorithms that allow for efficient probabilistic decoding [15] of trellis based encoders. The two most popular are the

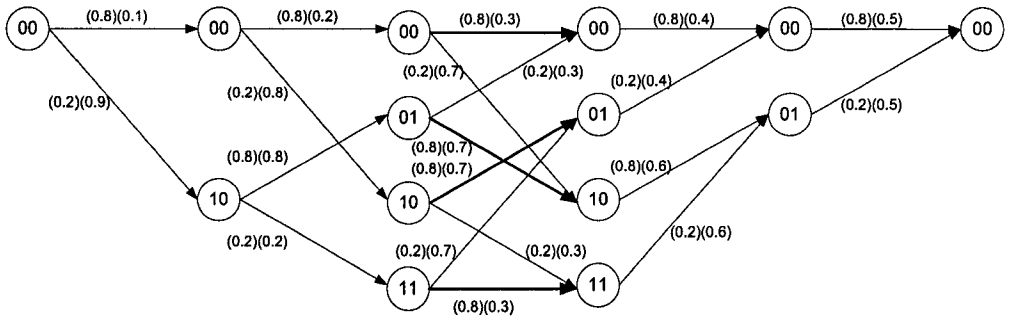

Fig. 2.12 A 5 stage trellis with associated transition metrics and transitions in the third stage that result from $X_3 = 0$ highlighted.

Viterbi Algorithm (VA) [5] and the BCJR algorithm (named after its authors Bahl, Cocke, Jelinek and Raviv) [2]. The VA algorithm computes the most likely transmitted packet $\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} P[\mathbf{x}|\text{all observations}]$, while the BCJR algorithm computes the *a posteriori* probability of each message bit $\Lambda_{j,i} = P[X_i = j|\text{all observations}]$.

In this work, only the BCJR algorithm is of relevance and it is best illustrated through the following example. Let $\mathbf{r}$ denote a vector of independent probabilities on the parity sequence bits available at the decoder, where $\mathbf{r}_{i,j} = P[U_j = i|\text{all observations}]$. Consider a binary source with $P[X_i = 0] = 0.8$ and the receiver has that

$$\mathbf{r} = \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.5 \\ 0.9 & 0.8 & 0.7 & 0.6 & 0.5 \end{bmatrix}. \tag{2.54}$$

Fig. 2.12 illustrates the complete trellis with $N = 5$ stages and associated edge *transition metrics*. A node $s_{i,j}$ is indexed by its stage i and state j while the edge metric between s_{i,j_1} and s_{i+1,j_2} will be denoted as $\gamma_i(j_1, j_2)$. In the event that no transition is possible, the metric is 0. A *transition metric* is a weight associated with each edge that is proportional to the probability of the transition.

The conditional probability $P[\mathbf{x}|\mathbf{r}]$ can be computed as

$$P[\mathbf{X} = \mathbf{x}'|\mathbf{r}] = \frac{P[\mathbf{r}|\mathbf{X} = \mathbf{x}']P[\mathbf{X} = \mathbf{x}']}{\sum_{\mathbf{x}} P[\mathbf{r}|\mathbf{X} = \mathbf{x}]P[\mathbf{X} = \mathbf{x}]} \tag{2.55}$$

$$= \frac{\prod_{k=1}^{N} P[\mathbf{r}|X_k = x'_k]P[X_k = x'_k]}{\sum_{\mathbf{x}} P[\mathbf{r}|\mathbf{X} = \mathbf{x}]P[\mathbf{X} = \mathbf{x}]}. \tag{2.56}$$

where the product $P[\mathbf{r}|X_k = x'_k]P[X_k = x'_k]$ is defined as the edge metric and these are shown in Fig. 2.12. The edge metric is simply the product of the probability of the received signal (conditioned on the expected signal given the transition in question) and the probability of the input symbol that produced the transition. It is convenient to denote by K the normalization constant present in the denominator of Eq. (2.56). Each message $\mathbf{x}$ corresponds to a path through the trellis for which there is an associated probability. The probability that the input bit was 0 in the third stage can then be expressed as

$$P[X_3 = 0|\mathbf{r}] = \sum_{\substack{\text{all paths } l \\ \text{s.t. } X_3=0}} P[l|\mathbf{r}]. \tag{2.57}$$

This may appear to be a hard problem at first, but in fact, due to the trellis structure, the BCJR algorithm provides an efficient means of calculation [2].

With each node $s_{i,j}$ in stage j, we associate a forward metric $\alpha_i(j)$ which corresponds to the sum along all paths between the initial node $s_{1,1}$ and node $s_{i,j}$ of the the product of the associated edge metrics along each path. Similarly, one can define a backward metric $\beta_i(j)$ between node $s_{i,j}$ and the last node $s_{N,1}$.

These metrics can easily be computed with the initial conditions $\alpha_1(1) = 1$, $\beta_N(1) = 1$ and the pair of recursive formulas

$$\alpha_{i+1}(j) = \sum_k \alpha_i(k)\gamma_i(k,j), \tag{2.58}$$

$$\beta_{i-1}(j) = \sum_k \beta_i(k)\gamma_{i-1}(j,k). \tag{2.59}$$

The probability that $X_i = a$ can then be expressed in the compact form

$$P[X_i = a|\mathbf{r}] = \frac{1}{K}\sum_k \sum_{k'} \alpha_i(k) \cdot \gamma_i(k,k') \cdot \beta_{i+1}(k') \cdot I_i(a,k,k'), \tag{2.60}$$

where $I_i(a,k,k')$ is the indicator function that is 1 if input symbol a causes a transition from state k in stage i to state k' in stage $i+1$ and is zero otherwise. For $X_3 = 0$ in the above example, one has that $P[X_3 = 0|\mathbf{r}] = 0.7852$. As a final note, Eqs. (2.58), (2.59) and (2.60) may be implemented efficiently as matrix multiplications.

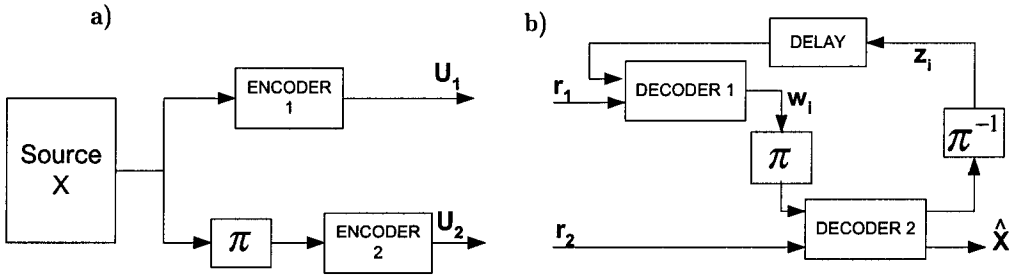

Fig. 2.13 The schematic block diagram of the turbo (a) encoder structure and (b) decoder structure.

2.8 Iterative Decoding and Turbo Codes

The most powerful channel coding techniques known today are based on concatenation and iterative decoding. Turbo codes and their extensions have reached performance close to the Shannon limit for the AWGN channel [39–41] and have been successfully applied to systems in wireless communication [42], digital recording [43], space communications [44, 45], etc. The objective of this section is to provide a sufficiently general framework for performing non-binary turbo-decoding as is done in Chapter 4.

A turbo code uses two or more constituent finite state machine encoders with trellis structure where the data to be encoded are interleaved prior to encoding. Fig. 2.13a shows a typical configuration of a turbo encoder.

Exact maximum likelihood decoding of a turbo code is difficult due to the interleaver. However, exact decoding of the constituent trellis codes is possible with the BCJR algorithm and a traditional turbo decoder consists of a serial concatenation of constituent decoders which exchange soft information (see Fig. 2.13b).

Each constituent decoder evaluates a logarithmic array of *a posteriori* probabilities. For example [46], at iteration i, the first constituent decoder may evaluate

$$\Lambda_1(\mathbf{x}|\mathbf{r}_1, \mathbf{z}_{i-1}) = \begin{bmatrix} \log P[x_1 = a_1|\mathbf{r}_1, \mathbf{z}_{i-1}] & \dots & \log P[x_N = a_1|\mathbf{r}_1, \mathbf{z}_{i-1}] \\ \vdots & \ddots & \vdots \\ \log P[x_1 = a_M|\mathbf{r}_1, \mathbf{z}_{i-1}] & \dots & \log P[x_N = a_M|\mathbf{r}_1, \mathbf{z}_{i-1}] \end{bmatrix}, \quad (2.61)$$

with the BCJR algorithm where the extrinsic information in $\mathbf{z}$ and $\mathbf{w}$ is treated as an

independent coordinate-wise measurement of the message $\mathbf{x}$ such that

$$P[\mathbf{x}|\mathbf{r},\mathbf{z}] = K_1 P[\mathbf{x}|\mathbf{r}] \prod_i \exp(z_{x_i,i}) \tag{2.62}$$

$$P[\mathbf{x}|\mathbf{r},\mathbf{w}] = K_2 P[\mathbf{x}|\mathbf{r}] \prod_i \exp(w_{x_i,i}), \tag{2.63}$$

where K_1 and K_2 are normalization constants. In practice, if an edge transition metric $\gamma_i(j_1, j_2)$ produces output symbol a_k, the extrinsic information is included in the BCJR algorithm by multiplying the corresponding edge transition metric by either $exp(z_{a_k,i})$ or $exp(w_{a_k,i})$. The extrinsic information may be evaluated according to the following pair of recursion equations

$$\mathbf{w}_i = \Lambda_1(\mathbf{x}|\mathbf{r}_1, \mathbf{z}_{i-1}) - \mathbf{z}_{i-1} - \mathbf{l}^{prior} \tag{2.64}$$

$$\pi(\mathbf{z}_i) = \Lambda_2(\mathbf{x}|\mathbf{r}_2, \pi(\mathbf{w}_i)) - \pi(\mathbf{w}_i) - \pi(\mathbf{l}^{prior}), \tag{2.65}$$

where $\mathbf{l}^{prior}$ is a logarithmic array of *a priori* probabilities $l_{j,i}^{prior} = P[X_i = a_j]$. Intuitively, the extrinsic information that is generated by a constituent decoder is a measure of the relative change between the input and output probability estimates on the message symbols.

In practice, the BCJR algorithm is sometimes too difficult to utilize in iterative decoders and is numerically unstable. This is because it involves a mix of non-linear functions and the numerical representation of probabilities [47]. Log domain implementations of the BCJR algorithm have been proposed to resolve these numerical instabilities. Once the soft probabilities on the coded and message symbols have been determined, the BCJR algorithm only requires the operations of addition, multiplication and division. The latter two are trivial to perform in the log domain but the first requires some though. The Max-Log-MAP algorithm [48] approximates this operation by performing a simple maximization operation as shown in Eq. (2.66).

$$\ln(e^{\delta_1} + e^{\delta_2}) \approx \max(\delta_1, \delta_2). \tag{2.66}$$

Due to the approximation, the Max-Log-MAP algorithm is suboptimal. This issue is addressed by the Log-MAP algorithm [47] which includes a correction term to Eq. (2.66):

$$\ln(e^{\delta_1} + e^{\delta_2}) = \max(\delta_1, \delta_2) + \ln(1 + e^{|\delta_2 - \delta_1|}). \tag{2.67}$$

The correction term only depends on the absolute value of the difference between δ_1 and δ_2: the correction term may be implemented as a one-dimensional look up table. It is reported in [47] that by quantizing the correction term to 8 stored values with $|\delta_2 - \delta_1|$ ranging from 0 to 5, no noticeable difference is observed between exact iterative decoding with the BCJR algorithm and the Log-MAP algorithm.

2.9 Published Work

A review of existing coding techniques for the various information theoretic problems outlined above is presented in this section. For the case of data compression, the techniques presented here have been established for several decades. For noise robust data compression, the Slepian-Wolf problem and the Wyner-Ziv problem, many of the techniques presented here have only appeared recently during the production of this thesis.

2.9.1 Data Compression

Historically, data compression has been of great interest and many of the coding results were obtained within 30 years of publication of Shannon's mathematical theory of communication. This section investigates Huffman codes, Lempel-Ziv coding and some results on data compression in the presence of noise. Note that both Huffman and Lempel-Ziv codes are variable length coding coding techniques in the sense that the number of output symbols is not fixed beforehand.

Huffman Coding

A traditional approach to source coding has been the mapping of a fixed length of source symbols to a variable length of coded symbols.

More formally, a source code C for a random variable X is a mapping from $\mathcal{X}$ to a set $\mathcal{D}$ consisting of finite length strings from a D-ary alphabet [27]. In this work, only binary alphabets ($D = 2$) will be considered though all the results are known to generalize to alphabets of arbitrary size. It is convenient to denote the codeword in $\mathcal{D}$ corresponding to $x \in \mathcal{X}$ as $C(x) \in \mathcal{D}$ and the length of the codeword as $l(x)$. One then has the obvious definition [27],

Definition 2.9.1 *The expected length $L(C)$ of a source code C for a random variable X is $L(C) = E[l(X)] = \sum_{x \in \mathcal{X}} p_X(x) l(x)$.*

Typically, a source with outputs $x_1, x_2, ..., x_n$ is compressed by concatenating the corresponding codewords to obtain $C(x_1, x_2, ..., x_n) = C(x_1)C(x_2)...C(x_n)$. Of particular interest is the ability to decode the concatenated code "on the fly" when the decoder has only received a partial string. One has the following definition [27],

Definition 2.9.2 *A code is called a* prefix *code if no codeword is a prefix of any other codeword.*

As the end of a codeword can be recognized immediately, it can be shown that prefix codes can be decoded without requiring knowledge of future codewords. Huffman codes, originally discovered in 1952, are optimal *prefix* codes for a given distribution [49]. Here, optimal is meant in the sense that the expected length of the codewords is minimal.

The construction of a Huffman code is best illustrated graphically with a tree. Consider a quaternary source with alphabet $\mathcal{X} = \{1, 2, 3, 4\}$ with probabilities 0.1, 0.2, 0.3, 0.4. The algorithm is illustrated in Fig. 2.14. First, all the symbols are listed in a column with their respective probabilities. The two least likely probabilities are merged and the resulting probabilities are listed in the next column. Since a merger has been performed, the edges leading to the merger are labelled with different binary symbols. The merging operation is repeated until there is only one node with probability 1 remaining. The codeword corresponding to each quaternary symbols can be obtaining be reading the edge labels through the tree from its root to the respective leaf.

Here, the average codeword length is 1.9 bits/symbol as compared to the theoretical limit $H(X) = 1.8464$. One can further improve performance by designing codes for pairs or n-tuples of input symbols. It can be shown [27] that such a method approaches the entropy H(X) asymptotically. As a final note, the encoder requires explicit knowledge of the source statistics.

Lempel-Ziv Coding

The Lempel-Ziv algorithm [50, 51], so named after its authors, was discovered in 1977. As opposed to Huffman codes, a variable length of input symbols is encoded into a variable length of coded symbols. Furthermore, the algorithm is *universal*: it does not require

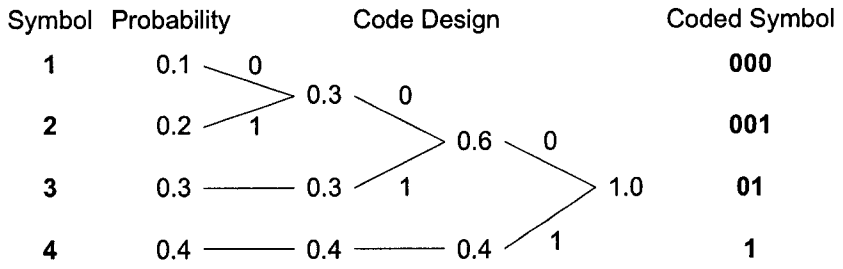

Fig. 2.14 The design of a Huffman code.

explicit knowledge of the source statistics. This is possible because the algorithm observes the outputs sequences of the source and assigns small indices to them. As the observed source sequences get longer, they tend to resemble typical sequences. The algorithm is as follows:

Parsing Parse the sequential outputs of the source into the shortest possible strings that have not yet appeared.

Encoding Since each string is the shortest possible that has not yet appeared, it must differ in only the last symbol from a previous string. Encode the new parsed string as a pair consisting of the index to the largest matching prefix and the additional symbol at the end.

The following example illustrates the algorithm. Consider the following binary output (generated by an i.i.d source with $P[X = 0] = 0.8$): 0101010010001000. The string is then parsed as 0,1,01,010,0100,01000. It is then encoded as the following sequence pairs: (0,0), (0,1), (1,1), (3,0), (4,0), (5,0). In practice, the index is usually allotted a fixed number of bits in the encoded stream. As the encoder operates on longer sequences of inputs, very long strings are recorded by short, fixed length indices.

It has been proved [27] that for stationary ergodic sources, the Lempel-Ziv algorithm performs arbitrarily close to the optimal rate achievable by any source code.

Noise Robustness

So far, it has always been assumed that the decoder had perfect knowledge of the encoder output. In practice, this information must always be transmitted over or stored on imperfect

physical media. Although powerful forward error correcting codes are often employed, recent research has been involved in the investigation of the noise robustness of source coding [52–57].

That this is possible at all is due to the fact that the source code does not achieve perfect compression and there exists a residual redundancy. The above works all seek to exploit the residual redundancy in variable length codes (VLCs).

Though very efficient in terms of compression (see Section 2.9.1), variable length codes are very sensitive to channel errors [52]. In fact, the more efficient the compression, the more sensitive the scheme is to noise. A single bit error in the compressed stream typically propagates errors at the decoder. Ideally, the decoder would evaluate the Maximum *A Posteriori* (MAP) packet $\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} P[\mathbf{x}|\mathbf{r}]$ where $\mathbf{r}$ is the (soft)-information available at the decoder. MAP decoding of variable length codes is computationally complex, so simpler, approximate techniques have been developed [52–54, 57]. Equally interesting is the result that some techniques assume knowledge of the number of source symbols compressed [53, 57] while others do not [55, 56]. In [56], using soft information resulted in a decrease by a factor of 10 in the frame error rate (FER) while in [52], efficient sub-optimal soft decoding performs very near optimal soft decoding and gains about 0.5 dB over hard decoding.

2.9.2 The Slepian-Wolf Problem

Recently published results on coding for the Slepian-Wolf problem may be found in the works of [58–69] These works may be classified into two categories: those that aim for zero-error coding (in [58–62, 65–67]), where compression is only possible in special cases (see Theorem 2.5.2 and Orlitsky [31]) and those that aim for near-lossless compression (in [63–65, 67–69] as well as the work presented in this thesis).

On the zero-error side, Pradhan and Ramchandran focus on a special case when the correlation between X and Y is specified as a prescribed maximal Hamming distance and explore the use of linear block codes in this setting [58]. For example, consider sources X and Y each producing 3-bit words whose correlation is specified by the fact that they differ in no more than 1 position. If the problem is treated so that Y is available at the decoder, then it would be wasteful for the X-encoder to differentiate between 000 and 111. The same can be said of the pairs $\{001, 110\}$, $\{010, 101\}$ and $\{100, 011\}$. It is sufficient for the X-encoder to specify in which of these sets the source output is.

Additional work on zero-error coding where the problem is treated as source coding with side-information at the decoder may be found in [60], [61] and [66]. In particular, [61] studies necessary and sufficient conditions for the existence of a lossless instantaneous code for binary sources and gives sufficient conditions for non-binary ones. They also show that the design in Kh et al. [60] is not optimal. In [66], Zhao and Effros propose a tree structured algorithm to construct a class of multiple access source codes. It has recently been shown by Koulgi et al. [62] that optimal design of zero-error codes with side information at the decoder is an NP-hard problem.

Near lossless coding techniques have been based on two different approaches: tree based variable length codes and turbo codes. Zhao and Effros extend their variable length codes to near-lossless coding and remove the restriction of treating the problem as a side-information scenario [65, 67]. That this is difficult for variable length codes follows from the fact that if the decoder decodes an incorrect symbol, it may loose synchronization with the compressed stream resulting in error propagation similar to that for traditional data compression. In the above works, the error propagation effect is carefully minimized by guaranteeing that the decoder will be able to re-synchronize after a number of symbols, regardless of any errors.

Independently and concurrently to the work presented in this thesis are the results by Garcia-Frias et al. in [63, 64]. There, each source is compressed separately by a pair of heavily punctured recursive systematic convolutional encoders operating in parallel as shown in Fig. 2.15. Since each source is encoded at the same rate $R_X = R_Y$, most of the achievable region for the Slepian-Wolf problem cannot be realized with this approach. Decoding is done iteratively in a similar fashion as outlined in Section 2.8 for turbo codes with the exception that once decoding of one source is finished, it is used to assist the decoding of the second source. This process is also repeated iteratively as illustrated in Fig. 2.16. This work has recently been extended to non-binary sources by converting the non-binary symbols into fixed length binary equivalents [69]. Also, a similar approach to the Slepian-Wolf problem may be found in [68] where the problem is treated as a side-information problem. Source Y is encoded with rate $R_Y = H(Y)$ and the encoder for X attempts to compress the source as closely as possible to $H(X|Y)$.

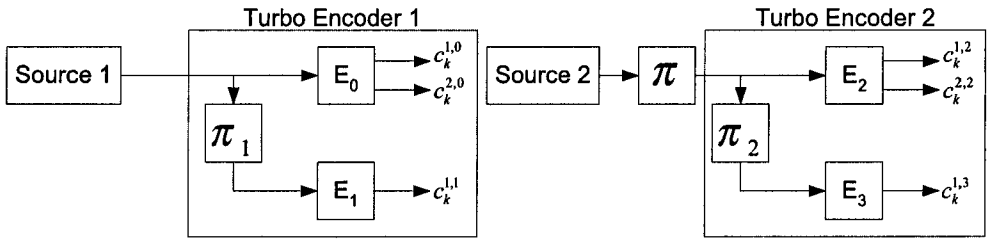

Fig. 2.15 Encoder structures utilized by Garcia-Frias et al.

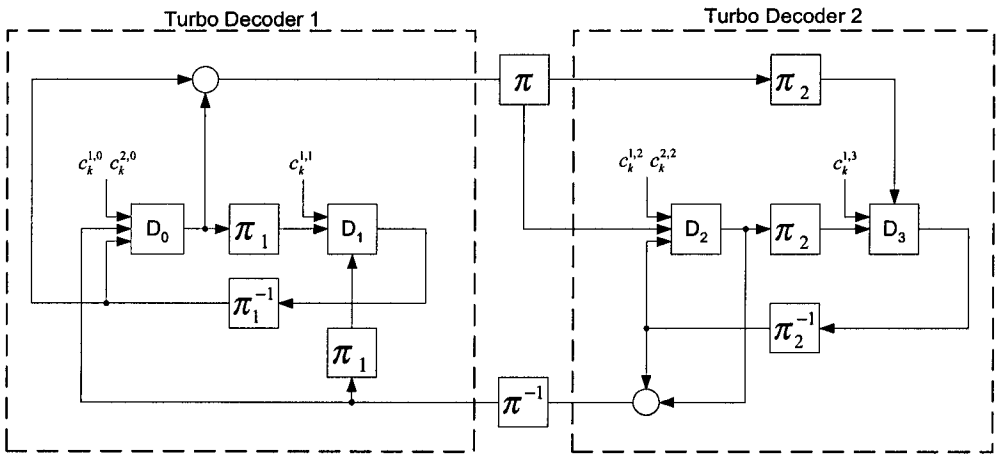

Fig. 2.16 Decoder structures utilized by Garcia-Frias et al.

2.9.3 The Wyner-Ziv Problem

Published coding results for the Wyner-Ziv problem may be found in [11, 58, 70–74].

In [70, 71], a "constructive" approach is taken to the Wyner-Ziv problem for the cases of binary symmetric and correlated Gaussian sources. In the case of correlated binary symmetric sources, it is shown how to partition parity check matrices to construct a code that can achieve the Wyner-Ziv rate distortion function. A critical assumption of the argument is the ability to perform typical decoding of parity check matrices. As an example of the latter, consider a binary source with bias $P[X = 0] = p$. It is assumed that one can find an $m \times n$ binary matrix H and decoding function f such that $f(H\mathbf{X}^T) = \mathbf{X}$ for most realizations of $\mathbf{X}$ provided $m/n > h(p)$ and n is taken sufficiently large. Although theoretically feasible, such codes have high (exponential) complexity in general and are not practical.

For the correlated Gaussian sources X and Y, the construction involves lattice codes [37] (a lattice Λ is given by the set of elements $\Lambda = \{i_1\mathbf{b}_1 + \cdots + i_N\mathbf{b}_N\}$ with $i_1, ..., i_N \in \mathbb{N}$ and $\mathbf{b}_1, ..., \mathbf{b}_N \in \mathbb{R}^n$). A pair of lattices Λ_1, Λ_2 are "nested" so that $\Lambda_1 \subset \Lambda_2$. To quantize $\mathbf{X}$, the nearest $\lambda_1 \in \Lambda_1$ and $\lambda_2 \in \Lambda_2$ to $\mathbf{X}$ are first determined. With the associated λ_1, there is a finite list of nearest neighbors in Λ_2. The encoder simply transmits the index of λ_2 from this list. At the decoder side, λ_1 must first be determined in order to recover λ_2 from the index. This is accomplished with the side-information $\mathbf{Y}$. If the minimum distance between lattice points of Λ_1 is sufficiently large, this can be achieved with low-probability of error. It was shown that for high correlation SNR, the distortion can be made close to the Wyner-Ziv rate-distortion function, though no actual construction of nested codes was proposed. The work has been extended in [74] where the restriction on high correlation has been removed, though no explicit design of lattices was proposed.

Servetto [11] later answered this challenge by providing a class of lattice codes for which it was shown that as n is made large and a high-degree of correlation between X and Y is assumed, they arbitrarily approach the Wyner-Ziv function irrespective of the code rate. Simulation results for specific lattices with $n = 2, 8, 24$ show good performance over a wide range of code rates and the E_8 lattice performs as close as 1.5dB from the rate-distortion curve. These results however, are for a strong correlation SNR (see Section 2.6) with $\sigma_X^2/\sigma_U^2 = 100$.

A different approach to the problem may be found in [58], where the proposed method has been based on quantizing X and coding for the related Slepian-Wolf problem with a trellis structured encoder. The design of embedded trellis encoders for source coding with side information was investigated in [73] where a coding gain of 1 dB over the results of [58] was observed.

It is interesting to note that a slightly different network problem is investigated in [59, 75], where *both* X and Y must be encoded with lattice codes for transmission to a common decoder with fidelity criterion.

Chapter 3

Code Design

This chapter presents the design of the FSM encoders used in the parallel concatenated codes. As opposed to the vast majority of the existing literature on turbo codes, both trellises are designed *jointly* and are non-binary, non-linear, non-systematic and non-symmetric.

3.1 Initial Considerations

The constituent encoders used in the concatenated framework perform data compression and produce only n output p-ary symbols for every k input p-ary symbols where $n < k$. Since we perform fixed-length to fixed-length data compression, several encoder input sequences must map into a single output sequence by the pigeonhole principle. Instead of achieving this by heavily puncturing the output of a (recursive) convolutional encoder, e.g., to achieve $k = 3$ and $n = 1$, it was preferred to custom-design time-invariant non-linear Finite State Machine (FSM) encoders that naturally produce fewer outputs than inputs. Moderate puncturing can then be used to achieve a wider range of rates. A couple of design rules, described in the following paragraphs, were chosen to guarantee good performance of the constructed FSM encoders. Fig 3.1 graphically illustrates the rules.

First, by choosing trellis structures with at least p^k states, one eliminates the need for parallel edges (see Fig. 3.1a). Hence, no two different k-symbol input sequences drive the encoder from the same current state to the same destination state. The construction in Theorem 3.2.1 will be concerned with the case of exactly p^k states while that of Theorem 3.3.2 will deal with the case of more states.

Second, to *introduce sufficient memory*, we avoid trellises where two or more branches

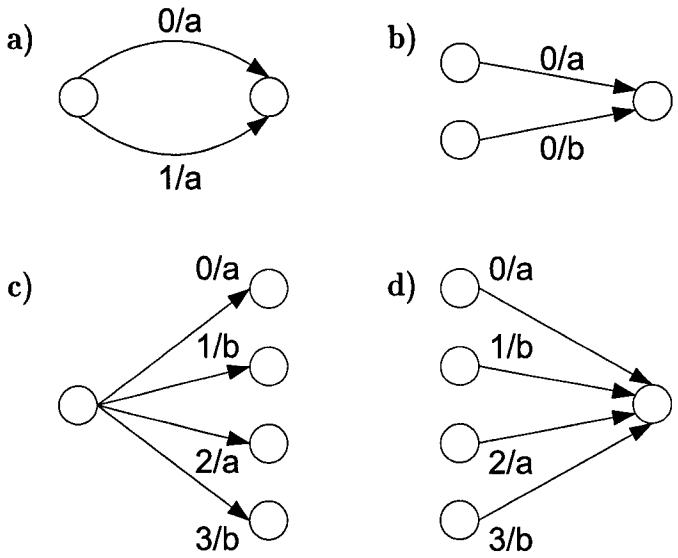

Fig. 3.1 A graphical illustration of the initial code design rules: (a) a forbidden trellis with parallel edges, (b) a forbidden trellis where two transitions with the same input edge label merge, (c) a good trellis with uniform spread of output edge labels leaving a state and (d) a good trellis with uniform spread of output edge labels merging into a state.

with the same k-symbol input edge label merge to the same state (see Fig. 3.1b). This effect would partially remove memory of the previous states and impede code performance. Traditional convolutional (non-recursive) codes are a good example of memory limited encoders and this is known to limit their performance in concatenated encoding schemes.

Finally, in order to *maximize output symbol usage*, it is also desirable that the set of all branches leaving from/merging into a particular state produces all output symbols equally often (see Figs. 3.1c and 3.1d).

Trellises that satisfy "equal spread" properties are often encountered in group codes [76]. In section 3.5, it will be shown that of the two trellis constructions presented here, the second always results in trellises with the *group trellis section* property. However, the first construction does not necessarily always have this property.

Recall that an FSM encoder is described by two matrices: an input state transition matrix whose (i,j)th entry corresponds to the k-symbol input sequence when the encoder makes a transition from state i to state j, and an output state transition matrix whose (i,j)th entry corresponds to the n-symbol output sequence for the described transition.

Limiting the discussion to the p^k states case for now, the particular requirements translate into the fact that these matrices have dimensions p^k by p^k with every entry unique in its column for the input state transition matrix. Since the input state transition matrix must also have every entry unique in its row (otherwise the same input sequence would lead to two different states), this matrix must be a Latin square. These have been extensively studied in combinatorial mathematics and can easily be constructed with the help of finite fields [77, 78]. Appendix B.1 reviews the properties of Latin squares that are relevant in this work and these are summarized below, without proof:

Two matrices $A_{i,j} = ai + j$ and $B_{i,j} = bi + j$ such that $a, b, i, j \in GF(p^k)$ and $a \neq b$ have many interesting properties, summarized in **[L1]** and **[L2]**.

[L1] Each entry is unique in its row and column.

[L2] The ordered pair $(A_{i,j}, B_{i,j})$ runs exactly once through all the elements of $GF(p^k) \times GF(p^k)$.

Matrices that satisfy condition **[L1]** are known as Latin squares. Pairs of matrices that satisfy both conditions **[L1]** and **[L2]** are known as Mutually Orthogonal Latin Squares (MOLS) and the properties outlined above will play an important role in the proofs to follow.

Since the output state transition matrix has only p^n different elements, $p^n < p^k$, it was impossible to require that each entry be unique in its row and column. The guidelines listed above imply that each output symbol occurs exactly p^{k-n} times per row and p^{k-n} times per column (such objects are called frequency squares [78]).

Based on the guidelines, the state transition matrices are constructed according to the following equations

$$M_{in1}(i_1, j_1) = \xi(ai_1 + j_1) \tag{3.1}$$

$$M_{out1}(i_1, j_1) = \chi(bi_1 + j_1) \tag{3.2}$$

$$M_{in2}(i_2, j_2) = \xi(ci_2 + j_2) \tag{3.3}$$

$$M_{out2}(i_2, j_2) = \vartheta(di_2 + j_2), \tag{3.4}$$

where $a, b, c, d, i_1, i_2, j_1, j_2 \in GF(p^k)$. These are illustrated graphically in Fig. 3.2. The mapping $\xi : GF(p^k) \rightarrow \{0, ..., p^k - 1\}$ is a bijection and uniquely assigns a given k-symbol

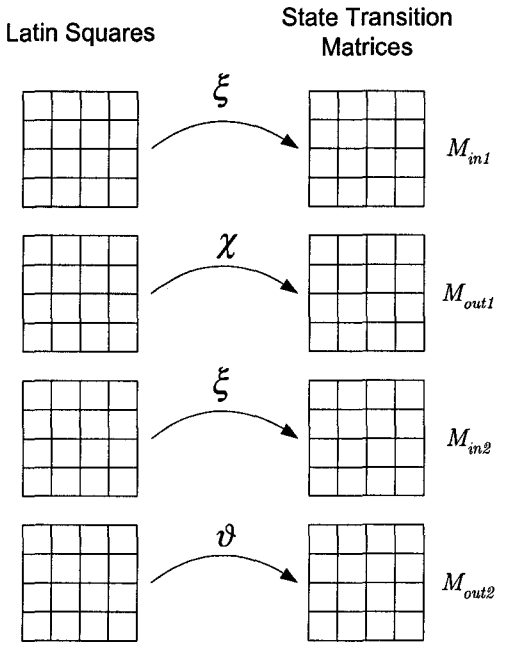

Fig. 3.2 Outline of the joint design of both FSM encoders. Mappings χ and ϑ will be designed jointly.

input sequence with a particular element of $GF(p^k)$. The many-to-one mappings χ, ϑ : $GF(p^k) \rightarrow \{0, ..., p^n - 1\}$ assign output symbols for given state transitions.

Definition 3.1.1 *A single finite state machine encoder constructed according to Eqs. (3.1) and (3.2) is said to be a Latin square based encoder. A pair of finite state machine encoders constructed according to Eqs. (3.1) - (3.4) is said to form a pair of Latin square based encoders.*

If the mappings χ and ϑ are chosen so that

$$|\chi^{-1}(v)| = p^{k-n} \tag{3.5}$$

$$|\vartheta^{-1}(v)| = p^{k-n}, \tag{3.6}$$

for all $v \in \{0, ..., p^n - 1\}$, then the state transition matrices for a Latin square based encoder will satisfy the outlined constraints, depicted in Fig. 3.1.

Theorem 3.1.1 *For a pair of Latin square based encoders obtained from a bijective* ξ :

$GF(p^k) \rightarrow \{0, ..., p^k - 1\}$ and whose construction satisfies the constraints in Eqs. (3.5) and (3.6), the following hold:

a) *The input state transition matrices are Latin squares.*

b) *The output state transition matrices have each entry appear p^{k-n} times per row and p^{k-n} times per column.*

c) *For any $a \neq b$, $c \neq d$, output symbol v and input symbol u, each encoder has exactly p^{k-n} states from which input symbol u generates output symbol v.*

Proof:
Part **a)** follows from property **[L1]** of Latin squares and the fact that ξ is a bijection.
Part **b)** follows from property **[L1]** and Eqs. (3.5) and (3.6).
Part **c)** follows from property **[L2]** and Eqs. (3.5) and (3.6).

□

Since both mappings χ and ϑ essentially transform a Latin square so that each entry occurs multiple times per row and column, this action is denoted as *collapsing a Latin square*. Given the conditions depicted in Fig. 3.1, there is still considerable freedom in the design of the three mappings χ, ϑ and ξ. It is desirable to narrow down the selection of these mappings by imposing additional constraints. For this, information theory can be used as a guide:

1. The entropy of the output of the FSM encoders must be maximized or else residual redundancy could be exploited to further compress the data. This translates into the requirement that the output of each encoder should be equally likely for a given distribution of the source.

2. There is no mutual information between the output streams of the FSM encoders, i.e. $I(\mathbf{r}_1; \mathbf{r}_2) = 0$, otherwise the shared information could be eliminated from one of the streams, thus improving the achieved compression.

Since these two requirements are difficult to satisfy at the message level, they are narrowed down to symbol level conditions which are used for the joint design of two good FSM encoders:

[C1] For a given i.i.d. source, the marginal distribution of the output of each FSM encoder is asymptotically uniform as the number of trellis stages is increased.

[C2] Given that it is known into which states encoders 1 and 2 were sent, there are exactly p^{k-2n} different inputs that may produce the observed output symbols.

The statement of condition **[C2]** requires some clarification. With each destination state j_1 of the first encoder and observed output v_1, there is a set $R_1(j_1, v_1)$ of inputs that may have occurred in the first encoder. Similarly, one has a set of $R_2(j_2, v_2)$ inputs that may drive the second encoder into state j_2 and produce output v_2. One requires that

$$|R_1(j_1, v_1)| = p^{k-n} \tag{3.7}$$

$$|R_2(j_2, v_2)| = p^{k-n} \tag{3.8}$$

$$|R_1(j_1, v_1) \cap R_2(j_2, v_2)| = p^{k-2n}, \tag{3.9}$$

for all j_1, j_2, v_1 and v_2. The core of each proof will be to show that it is possible to construct the requisite mappings χ, ϑ and ξ that guarantee properties **[C1]** and **[C2]**.

3.2 Design with p^k States

Theorem 3.2.1 *For any prime p and positive integers $k \geq 2n$, there exists a pair of Latin square based encoders with p^k states satisfying conditions* **[C1]** *and* **[C2]**.

Proof: The proof will be done by constructing suitable mappings for Eqs. (3.1) - (3.4) and Eqs. (3.5) and (3.6). In this case, we have a p^n-ary output symbol generated for each p^k-ary input symbol. The trellises have p^k states, i.e. the minimum needed to avoid parallel edges. The M_{in} and M_{out} matrices are of size $p^k \times p^k$ and $a, b, c, d, i_1, i_2, j_1, j_2 \in GF(p^k)$ while $\xi : GF(p^k) \rightarrow \{0, ..., p^k - 1\}$ is one-to-one and onto. Furthermore, choose $a \neq b$ and $c \neq d$.

First, the ϑ mapping shall be constructed. Galois fields $GF(p^k)$ are often denoted as $(k-1)$ order polynomials over a prime field of order p (commonly denoted F_p). More conveniently, one can simply think of this polynomial as a vector in F_p^k. Define the mapping ϑ by an arbitrary projection onto n coordinates of the vector space. Consequently, $|\vartheta^{-1}(v)| = p^{k-n}$ and Eq. (3.5) is satisfied.

To determine the mapping χ, for each $v \in \{0, ..., p^n - 1\}$ partition $\vartheta^{-1}(v)$ into sets $C_{v,0}, C_{v,1}, ..., C_{v,p^k-1}$ of size p^{k-2n}. Construct a family of sets according to $C_m = \bigcup_v C_{v,m}$. The mapping χ is then defined by the rule $\chi(\alpha) = m$ if $\alpha b^{-1} a c^{-1} d \in C_m$. It is clear that $|\chi^{-1}(v)| = p^{k-n}$ and therefore Eq. (3.6) is satisfied.

Part 1: Condition [C1]

A given input symbol can only produce one output symbol from a given state. One can then express the probability of the encoder generating an output symbol v in terms of the joint probability of the input symbol u and encoder state i pair as follows:

$$P(v) = \sum_{\substack{(u,i) \text{ that} \\ \text{produce } v}} P(i,u) \tag{3.10}$$

$$= \sum_{\substack{(u,i) \text{ that} \\ \text{produce } v}} P(i|u)P(u) \tag{3.11}$$

$$= \sum_{\substack{(u,i) \text{ that} \\ \text{produce } v}} P(i)P(u). \tag{3.12}$$

If the probability distribution of the states is uniform, i.e. $P(i) = p^{-k}$, then one obtains

$$P(v) = \sum_{\substack{(u,i) \text{ that} \\ \text{produce } v}} p^{-k} P(u) \tag{3.13}$$

$$= \sum_{\text{all inputs } u} P(u) p^{k-n} p^{-k} \tag{3.14}$$

$$= p^{-n}. \tag{3.15}$$

where Eq. (3.14) follows from Theorem 3.1.1c. Hence, to show that the marginal distribution of the coded symbols asymptotically approaches a uniform distribution, it will suffice to show that the probability distribution of the states tends to a uniform distribution. A proof of this may be found in Appendix B.2.

Part 2: Condition [C2]

It still needs to be shown that condition **[C2]** is satisfied. Each output symbol occurs p^{k-n} times per row and p^{k-n} times per column in the output state transition matrix. For encoder 1, the set of rows in which output symbol $v_1 \in \{0, ..., p^n - 1\}$ occurs in column

$j_1 \in GF(p^k)$ is given by

$$i_1 \in b^{-1}\chi^{-1}(v_1) - b^{-1}j_1. \tag{3.16}$$

The input symbol that generated this output is in row i_1 of column j_1 of the M_{in1} matrix as given by Eq. (3.1). One needs a method of determining in which row the input symbol occurs in column j_2 of M_{in2}. This can be found by equating Eq. (3.1) with Eq. (3.3) and since ξ is bijective:

$$\xi(ai_1 + j_1) = \xi(ci_2 + j_2) \tag{3.17}$$

$$ai_1 + j_1 = ci_2 + j_2 \tag{3.18}$$

$$i_2 = c^{-1}ai_1 + c^{-1}(j_1 - j_2). \tag{3.19}$$

If this is combined with Eq. (3.16), one obtains a set of rows i_2 corresponding to the locations of the inputs in column j_2 of the second encoder which result in output v_1 in column j_1 of the first encoder:

$$i_2 \in c^{-1}ab^{-1}\chi^{-1}(v_1) - c^{-1}ab^{-1}j_1 + c^{-1}(j_1 - j_2). \tag{3.20}$$

Eqs. (3.20) and (3.4) may be used to define a list $V_2(v_1, j_1, j_2)$ according to

$$V_2(v_1, j_1, j_2) = \vartheta\left[dc^{-1}ab^{-1}\chi^{-1}(v_1) + j_2(1 - dc^{-1}) + j_1(1 - ab^{-1})dc^{-1}\right], \tag{3.21}$$

which can be interpreted as *a list of all output symbols of the second encoder when entering state j_2 when the first encoder is entering state j_1 and generates output symbol v_1*. The requirement for good FSM design is that this list contains p^{k-2n} copies of each of the p^n output alphabet symbols for all combinations of $v_1 \in \{0, ..., p^n - 1\}$ and $j_1, j_2 \in GF(p^k)$. However, the effect of the j_1 and j_2 terms can be combined into a single term, j, which leads to the simplification,

$$V_2(v_1, j) = \vartheta\left[dc^{-1}ab^{-1}\chi^{-1}(v_1) + j\right]. \tag{3.22}$$

It will suffice to show that the list $V_2(v_1, j)$ contains p^{k-2n} copies of each of the p^n output alphabet symbols for all combinations of $v_1 \in \{0, ..., p^n - 1\}$ and $j \in GF(p^k)$. Now,

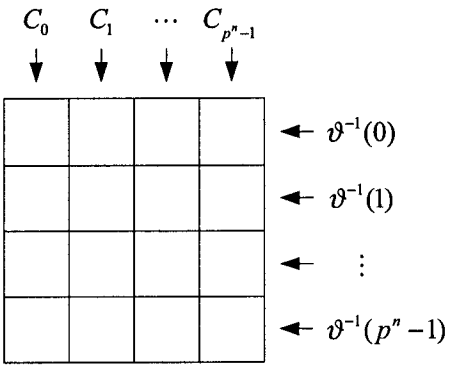

Fig. 3.3 A graphical representation of how the sets C_m and $\vartheta^{-1}(.)$ partition the input symbols when $j = 0$.

$V_2(v_1, j) = \vartheta\left[dc^{-1}ab^{-1}\chi^{-1}(v_1) + j\right] = \vartheta\left[C_{v_1} + j\right]$ which by construction of the mapping ϑ contains each output symbol p^{k-2n} times for any value of j. Fig. 3.3 illustrates the symbol resolution for the case $j = 0$.

□

The completion of the proof presented a constructive solution for the mappings χ and ϑ but did not state any condition on ξ. In particular, this means one is free to assign any k length p-ary sequence to any element of $GF(p^k)$ provided the mapping is bijective. In practice, the mapping is not so arbitrary. From condition **[C2]**, one has pairs of k-length symbols that can only be resolved indirectly by the memory of the code. Intuitively, it is desirable to maximize the Hamming distance of these k length symbols to aid the memory of the code. Usually this worked well (e.g., the rate 4/5 code in Appendix A) although in some cases, such as the rate 2/3 code in Appendix A, maximizing the Hamming distance did not result in the best observed performance.

3.3 Design with More than p^k States

The second case to consider is the situation when the FSM encoders have more than p^k states. The code is still constructed according to Eqs. (3.1)-(3.4) with the slight modifications that $a, b, c, d, i_1, i_2, j_1, j_2 \in GF(p^l)$ (with $l \geq k$) and the mapping ξ has a dummy output to denote impossible state transitions. In particular, we have that $\xi : GF(p^l) \rightarrow \{\phi, 0, ..., p^k - 1\}$ with $l \geq k$ where ξ is onto and each $u \in \{0, ..., p^k - 1\}$ is the image of a unique element $w \in GF(p^l)$. In essence, one is only concerned with those

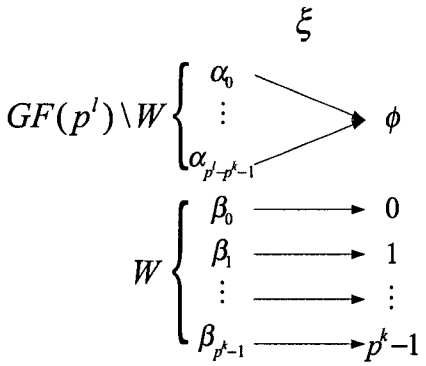

Fig. 3.4 A graphical illustration of the ξ mapping. The restriction of ξ to W is a bijection between W and $\{0, ..., p^k - 1\}$ while all the other elements $GF(p^l)\backslash W$ are mapped to the dummy symbol ϕ.

elements $w \in W \subset GF(p^l)$ such that $\xi(w) \in \{0, ..., p^k - 1\}$, i.e. $W = \xi^{-1}(\{0, ..., p^k - 1\})$ (see Fig. 3.4).

Definition 3.3.1 *A mapping $\xi : GF(p^l) \rightarrow \{\phi, 0, ..., p^k - 1\}$ with $p^k \geq 2$ is said to be W-bijective if there is a set $W \subset GF(p^l)$ so that $|W| = p^k$ and ξ_W, the restriction of ξ to W, is a bijection between W and $\{0, ..., p^k - 1\}$.*

Under the constraints

$$|\vartheta^{-1}(v)| = p^{l-n} \tag{3.23}$$

$$|\chi^{-1}(v)| = p^{l-n} \tag{3.24}$$

$$|\vartheta^{-1}(v) \cap W| = p^{k-n} \tag{3.25}$$

$$|\chi^{-1}(v) \cap W| = p^{k-n}, \tag{3.26}$$

it is easy to verify that results analogous to Theorem 3.1.1 still apply:

Theorem 3.3.1 *For a pair of Latin square based encoders obtained with a W-bijective $\xi : GF(p^l) \rightarrow \{\phi, 0, ..., p^k - 1\}$ and whose construction satisfies the constraints in Eqs. (3.23) - (3.26), the following hold:*

a) *Each symbol in $\{0, ..., p^k - 1\}$ appears once per row and once per column in the input state transition matrices.*

b) *In the transitions that are valid, the output state transition matrices have each entry appear p^{k-n} times per row and p^{k-n} times per column.*

c) *For any $a \neq b$, $c \neq d$, output symbol v and input symbol u, each encoder has exactly p^{l-n} states from which input symbol u generates output symbol v.*

Proof:
Part **a)** follows directly from property [**L1**] of Latin squares and the fact that each $u \in \{0, ..., p^k - 1\}$ is the image of a unique element in $GF(p^l)$.
Part **b)** is a consequence of Eqs. (3.25) and (3.26) and property [**L1**] .
Part **c)** is a consequence of Eqs. (3.23) and (3.24) and property [**L2**] .

□

The constructive proof of the existence of Latin square based FSM encoders that meet conditions [**C1**] and [**C2**] with more than p^k states will proceed in a similar fashion as the proof for encoders with p^k states. In particular, condition [**C1**] will be proved by showing that the probability distribution of the output symbols tends to a uniform distribution and [**C2**] will be shown by constructing the requisite mappings. Key to the proof of condition [**C1**] was that none of the transition probabilities $p_{i,j}$ were zero. With more than p^k states, this need not be true anymore. It will be necessary to investigate under what conditions are the states of the FSMs connected in the sense that one can get from any state to any other state given a suitable input sequence. The following lemma is a sufficient condition:

Lemma 3.3.1 *If $-a$ and $-c$ are both primitive elements of $GF(p^l)$, then the trellises of the Latin square based encoders are connected for any ξ that is W-bijective.*

Proof: Consider an input state transition matrix derived from a Latin square with parameter $a \in GF(p^l)$. Consequently, given an input symbol $u = \xi(w) \in \{0, ..., p^k - 1\}$ and current state $X_n = i$, the next state $X_{n+1} = j$ satisfies by Eq. (3.1):

$$\xi(aX_n + X_{n+1}) = \xi(w) \tag{3.27}$$

$$X_{n+1} = w - aX_n. \tag{3.28}$$

It will be shown that for each given input u, there is a state X_0 such that if the input is kept fixed, the encoder will cycle through $p^l - 1$ states (i.e. all but one). By Eq. (3.28),

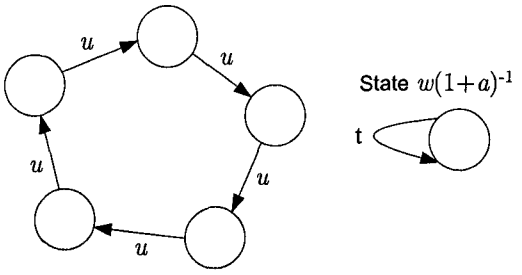

Fig. 3.5 An illustration of the state cycle generated when the input $u = \xi(w)$ is kept fixed.

the state at time n is

$$X_n = w(1-(-a)^n)(1+a)^{-1} + (-a)^n X_0. \tag{3.29}$$

The cycle repeats the first time $X_0 = X_n$. First, consider $w \neq 0$ and choose $X_0 = 0$. Then, it is clear that with $-a$ a primitive element, $X_0 = X_n$ for the first time when $n = p^l - 1$. Second, consider $w = 0$ and choose $X_0 = 1$. Since the sequence of states is then a geometric series in $(-a)$, $X_0 = X_n$ for the first time when $n = p^l - 1$.

Now, the state that is not present in the maximum length cycle is the one that gets sent to itself by the input symbol (i.e. it is stationary with respect to the input $u = \xi(w)$) and from Eq. (3.28) is given as $w(1+a)^{-1}$ (see Fig. 3.5). However, for a different input (and there must exist at least 2 different possible inputs into the FSM), a second cycle is formed that includes the original stationary $w(1+a)^{-1}$ state. By the pigeonhole principle, both cycles must overlap, hence all states can be reached.

□

Theorem 3.3.2 *For any positive integers $l \geq k \geq 2n$, there exists a pair of Latin square based encoders with p^l states for which conditions* **[C1]** *and* **[C2]** *hold.*

Proof: For the case $l \geq k$, choose any $\xi : GF(p^l) \rightarrow \{\phi, 0, ..., p^k - 1\}$ that is W-bijective. Choose $-a$ and $-c$ to be primitive elements and any $b \neq a$ and $d \neq c$.

Construct three projections P_ϑ, P_χ and P_U that map elements in $GF(p^k)$ in vector form onto n, n and $l-k$ different coordinates respectively. Construct $\vartheta(\alpha) = P_\vartheta(\alpha)$, $\chi(\alpha) = P_\chi(cd^{-1}ba^{-1}\alpha)$, $U = P_U^{-1}(0)$ and $W = cd^{-1}U$. Since we deal with projections, it is clear that Eqs. (3.23)-(3.26) are satisfied.

Part 1: Condition [C1]

Similar to the proof of Theorem 3.2.1, part 1, it is sufficient to show that the probability density of the states approaches the uniform distribution due to Theorem 3.3.1c. A proof of this may be found in Appendix B.3.

Part 2: Condition [C2]

Similarly to part 2 of Theorem 3.2.1, one obtains that the output v_1 in column j_1 occurs in the rows i_1 such that

$$i_1 \in b^{-1}\chi^{-1}(v_1) - b^{-1}j_1. \tag{3.30}$$

Also, both input state transition matrices may be related by equating Eq. (3.1) with Eq. (3.3). Furthermore, if one is only interested in the relation for valid input symbols (i.e. not ϕ), then

$$\xi_W(ci_2 + j_2) = \xi_W(ai_1 + j_1). \tag{3.31}$$

By combining Eq. (3.31) with Eq. (3.30) and the fact that ξ_W is a bijection, one obtains

$$ci_2 + j_2 \in W \cap (ab^{-1}\chi^{-1}(v_1) + (1 - ab^{-1})j_1) \tag{3.32}$$

A list of outputs $V_2(v_1, j_1, j_2)$ of the second encoder is then:

$$V_2(v_1, j_1, j_2) = \vartheta\left[\{dc^{-1}W \cap dc^{-1}(ab^{-1}\chi^{-1}(v_1) + (1 - ab^{-1})j_1)\} + (1 - dc^{-1})j_2\right] \tag{3.33}$$

$$= \vartheta\left[\{U \cap (dc^{-1}ab^{-1}\chi^{-1}(v_1) + dc^{-1}(1 - ab^{-1})j_1)\} + (1 - dc^{-1})j_2\right]. \tag{3.34}$$

Similarly to Theorem 3.2.1, one is seeking to demonstrate that mappings ϑ and χ result in lists $V_2(v_1, j_1, j_2)$ containaining p^{k-2n} copies of each of the p^n output alphabet symbols for all values of $v_1 \in \{0, ..., p^n - 1\}$ and $j_1, j_2 \in GF(p^l)$. Due to the projective nature of the mappings χ, ϑ and P_U, it is easy to verify that this is so with the proposed construction.

□

A few points can be made about Theorem 3.3.2. First, the requirement that the input state transition matrix is based on the negative of a primitive element in part 1 can be relaxed. All that is really required is that the states of the FSM are connected. This is always true if the above construction is utilized with $l = k$ and almost always true with

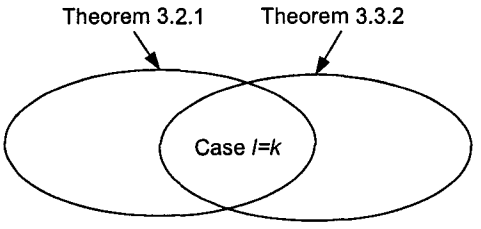

Fig. 3.6 The relationship between the constructions in Theorems 3.2.1 and 3.3.2.

$l > k$.

The second comment is about the relation between the constructions in theorems 3.2.1 and 3.3.2. If we constrain ourselves to the case $l = k$ in Theorem 3.3.2, then the construction presented there is a special case of that in Theorem 3.2.1 (see Fig. 3.6). In particular, while the mapping χ is a projection in Theorem 3.3.2, it was of a much more general form in Theorem 3.2.1. One advantage of the projection mapping is that all the input symbols are resolved the same way, regardless of the destination states of the encoders. In particular, if $u \in R_1(j_1, v_1) \cap R_2(j_2, v_2)$ and $u \in R_1(j_1', v_1') \cap R_2(j_2', v_2')$, then in fact $R_1(j_1, v_1) \cap R_2(j_2, v_2) = R_1(j_1', v_1') \cap R_2(j_2', v_2')$. The next section gives a design example where this is illustrated.

3.4 Design Example

Table 3.1 illustrates the construction of a code that generates 2 output bits for every 4 that are input. The code designed here was simulated in Chapter 4 and may be found in Appendix A.2. This code was generated with parameters $\{a, b, c, d\} = \{g^3, g^1, g^{10}, g^{14}\}$ where g is a primitive element of $GF(2^4)$.

The first two columns list the elements of the finite field in vector notation (left) and exponential notation (right). The third and fifth columns are the rightmost and leftmost coordinates of the vector representation. The fourth and sixth columns are obtained by multiplying the field elements by $cd^{-1}ba^{-1}$ and cd^{-1} respectively. The pairs shown in columns 2 and 3 give the ϑ mapping while columns 4 and 5 specify the χ mapping.

There remains the issue of deciding which k-bit symbols to assign to each element of the field. As stated earlier, we choose to maximize the Hamming distance between those inputs that cannot be resolved directly from the outputs they create. These can be found

Table 3.1 Code design of rate 2/4 trellises. This table constructs the ϑ and χ mappings.

Field Element					W equivalent
Vector g^x	x	$\vartheta()$	$g^x cd^{-1}ba^{-1}$	$\chi()$	$g^x cd^{-1}$
0 0 0 0	$-\infty$	0	$-\infty$	0	$-\infty$
0 0 0 1	3	1	12	0	14
0 0 1 0	2	0	11	0	13
0 0 1 1	6	1	0	0	2
0 1 0 0	1	0	10	0	12
0 1 0 1	9	1	3	0	5
0 1 1 0	5	0	14	0	1
0 1 1 1	11	1	5	0	7
1 0 0 0	0	0	9	1	11
1 0 0 1	14	1	8	1	10
1 0 1 0	8	0	2	1	4
1 0 1 1	13	1	7	1	9
1 1 0 0	4	0	13	1	0
1 1 0 1	7	1	1	1	3
1 1 1 0	10	0	4	1	6
1 1 1 1	12	1	5	1	8

by taking the entries in the last column (the W equivalent column) that occur where both ϑ and χ agree. Then, one obtains the following four sets $\{g^{-\infty}, g^{13}, g^{12}, g^1\}$, $\{g^{11}, g^4, g^0, g^6\}$, $\{g^{14}, g^2, g^5, g^7\}$, and $\{g^{10}, g^9, g^3, g^8\}$. These are assigned bit patterns according to a $(4, 2, 2)$ block code. Table 3.2 provides the final mapping ξ.

With the three mappings ξ, ϑ and χ as well as $\{a, b, c, d\} = \{g^3, g^1, g^{10}, g^{14}\}$ where g is a primitive element of $GF(2^4)$, one may now proceed to construct the trellises according to Eqs. (3.1)-(3.4). The trellises may be found in Appendix A.2 for the rate 2/4 code designed here and in Appendix A for the other designed trellises.

3.5 Group Structure

The group structure of the designed trellis sections will be briefly investigated. It will be shown that the trellis construction presented in Theorem 3.3.2 produces a *group trellis section.* Before proceeding to demonstrate this, a few definitions are required. The notation of [79] is followed with the sole difference that here, the input and output edge labels are

Table 3.2 Design of the ξ mapping for the rate 2/4 code.

	W equivalent	Bit Pattern
$\vartheta() = 0, \chi() = 0$	$-\infty$	0 0 0 0
	13	1 1 0 0
	12	0 1 1 0
	1	1 0 1 0
$\vartheta() = 0, \chi() = 1$	11	1 1 1 1
	4	0 0 1 1
	0	1 0 0 1
	6	0 1 0 1
$\vartheta() = 1, \chi() = 0$	14	0 0 0 1
	2	1 1 0 1
	5	0 1 1 1
	7	1 0 1 1
$\vartheta() = 1, \chi() = 1$	10	1 1 1 0
	9	0 0 1 0
	3	1 0 0 0
	8	0 1 0 0

separated for clarity. The group trellis property will be proved for the first trellis in the construction of Theorem 3.3.2. The result follows analogously for the second one. Since all groups in this section are abelian, the + operator is employed.

Recall that a trellis section is a five-tuple $X = (G, S, G', S', B)$, where G and G' are the input and output alphabets respectively, S and S' are the left and right states respectively, and B is the set of branches (see section 2.7). It is said that X is a *group trellis section* if S, S', G and G' are groups and B is a subgroup of the direct product $S \times S' \times G \times G'$ [79].

Theorem 3.5.1 *The trellis sections constructed in the proof of Theorem 3.3.2 are group trellis sections.*

Proof: First, it will be shown that S, S', G and G' each have group structure. Clearly, since $i \in S = GF(p^l)$ and $j \in S' = GF(p^l)$, both S and S' have group structure.

Recall that the mapping χ may be described as $\chi(\alpha) = P_\chi(cd^{-1}ba^{-1}\alpha)$, where P_χ is a projection mapping. Since the obvious relation $\chi(\alpha) + \chi(\beta) = \chi(\alpha + \beta)$ holds, χ is a homomorphism and the range $S' = \{0, .., p^n - 1\}$ of χ has group structure.

Finally, to demonstrate that G is a group, consider when $\xi(\alpha) \neq \phi$:

$$\begin{aligned}
\xi(\alpha) \neq \phi &\Leftrightarrow \alpha \in W && (3.35)\\
&\Leftrightarrow \alpha \in cd^{-1}U && (3.36)\\
&\Leftrightarrow \exists u \in U \text{ s.t. } \alpha = cd^{-1}u && (3.37)\\
&\Leftrightarrow \exists u \text{ s.t. } P_\xi(u) = 0 \text{ and } \alpha = cd^{-1}u && (3.38)\\
&\Leftrightarrow P_\xi(c^{-1}d\alpha) = 0. && (3.39)
\end{aligned}$$

.

Eqs. (3.35) and (3.39) imply that W is the kernel of a homomorphism and hence W is a group in its own right. Since the mapping ξ is a bijection between W and $G = \{0, ..., p^k - 1\}$, a group structure is induced onto G. We then have that if $\alpha, \beta \in W$, then $\xi(\alpha) + \xi(\beta) = \xi(\alpha + \beta)$.

It now follows that each of S, S', G and G' are groups and it remains to be shown that B is a subgroup under the direct product $S \times S' \times G \times G'$. In the construction of Theorem 3.3.2, the branches of the first trellis are given according to the formula,

$$B = \{(i, j, \xi(ai + j), \chi(bi + j)) : \xi(ai + j) \neq \phi\}. \tag{3.40}$$

Since $B \subseteq S \times S' \times G \times G'$ and the latter is a group, it will be sufficient to show that $\alpha, \beta \in B$ implies $\alpha + \beta \in B$ [80]. Let $\alpha, \beta \in B$ such that,

$$\alpha = (i_\alpha, j_\alpha, \xi(ai_\alpha + j_\alpha), \chi(bi_\alpha + j_\alpha)) \tag{3.41}$$

$$\beta = (i_\beta, j_\beta, \xi(ai_\beta + j_\beta), \chi(bi_\beta + j_\beta)). \tag{3.42}$$

Then,

$$\begin{aligned}
\alpha + \beta &= (i_\alpha + i_\beta, j_\alpha + j_\beta, \xi(ai_\alpha + j_\alpha) + \xi(ai_\beta + j_\beta), \chi(bi_\alpha + j_\alpha)) + \chi(bi_\beta + j_\beta)) && (3.43)\\
&= (i_\alpha + i_\beta, j_\alpha + j_\beta, \xi(a(i_\alpha + i_\beta) + (j_\alpha + j_\beta)), \chi(b(i_\alpha + i_\beta) + (j_\alpha + j_\beta))) && (3.44)\\
&\in B. && (3.45)
\end{aligned}$$

□

A few interesting points can be made about the group trellis structure of the codes presented here. The first is that all the codes employed in the simulations of Chapter 4 were constructed as described in Theorem 3.3.2 and therefore have group trellis structure. The actual trellises presented in Appendix A were optimized by constructing several hundred trellises according to Theorem 3.3.2 and choosing those which exhibited the best simulated performance for the Slepian-Wolf problem. It is important to note that some trellises exhibited poor performance and group trellis structure is not a sufficient condition for good performance.

Secondly, this section has so far remained silent as to the trellis structure of the construction in Theorem 3.2.1. In that construction, the mapping χ was of a much more general form than that in Theorem 3.3.2. Not all such mappings in Theorem 3.2.1 result in a homomorphism between $GF(p^k)$ and G'. Therefore, a group trellis structure is not a necessary condition for the equal spread properties of the output symbols in the trellis.

3.6 Extension to Rates Greater than 1

The Latin square based code design can easily be extended to rates greater than 1. Of course, in such a case, each k length input sequence would generate an n length output sequence with $k \leq n$. In other words, an input sequence is mapped to a unique output sequence and the symbol resolution criterion of [**C2**] is no longer relevant.

However, the concept of a uniform distribution on the output symbols can be applied to a trellis design [81]. Since condition [**C2**] was the key for a joint encoder design, this section will only be concerned with the design of a single trellis subject to the following conditions:

[**C1**] The marginal distribution of the output of each FSM encoder is asymptotically uniform as the number of trellis stages is increased.

[**C2'**] Each FSM encoder is a one-to-one mapping.

In the case of a trellis with rate greater than 1, the single trellis is described by a pair of input and output state transitions matrices constructed according to the familiar equations:

$$M_{in}(i,j) = \xi(ai + j) \tag{3.46}$$

$$M_{out}(i,j) = \chi(bi+j), \tag{3.47}$$

where $\xi : GF(p^n) \rightarrow \{\phi, 0, ..., p^k - 1\}$ with $k \leq n$ and $\chi : GF(p^n) \rightarrow \{0, ..., p^n - 1\}$. Now, let W be the pre-image under ξ of the non-dummy symbols, $W = \xi^{-1}(\{0, ..., p^k - 1\})$. Then, it is also required that the restriction of ξ to W, denoted ξ_W, and χ are both bijections.

Theorem 3.6.1 *For a Latin square based encoder with constraints on ξ and χ as outlined above, the following hold:*

a) *Each symbol in $\{0, ..., p^k - 1\}$ appears once per row and once per column in the input state transition matrix.*

b) *Each output symbol appears once per row and once per column in the output state transition matrix.*

c) *For any $a \neq b$, output symbol v and input symbol u, there is one and only one state from which input symbol u generates output symbol v.*

Proof:
Part **a)** follows directly from property **[L1]** of Latin squares, Eq. (3.46) and the fact that ξ_W is a bijection.
Part **b)** follows from property **[L1]** , Eq. (3.47) and the fact that χ is a bijection.
Part **c)** follows from property **[L2]** , Eqs. (3.5) and (3.6) and the fact that both ξ_W and χ are bijections.

□

Theorem 3.6.2 *For any integer $n \geq k$, there exists a finite state machine that satisfies conditions* **[C1]** *and* **[C2']**.

Proof:

Choose a to be the negative of a primitive element, any $b \neq a$, any bijection χ and any mapping $\xi : GF(p^n) \rightarrow \{0, ..., p^k - 1\}$ such that ξ is bijective.
Part 1: Condition [C1]

By Lemma 3.3.1, the trellis is connected. The result of Theorem B.3.1 (with n substituted for l) shows that the probability density of the states approaches a uniform distribution. The result may then be shown similar to Theorem 3.2.1 part 1.

Part 2: Condition [C2']

By theorem 3.6.1b, if the current state of the encoder is known, then knowledge of the output of the encoder will uniquely determine the next state of the encoder as well as the input that caused the transition. By induction, one may determine the unique input since it is assumed that the encoder was initialized to a known state.

□

3.7 Chapter Summary

This chapter has presented two constructions for pairs of finite state machine encoders. Trellis design rules in terms of an equal spread of input and output symbols were presented (see Fig. 3.1). Constructing trellises with these properties was shown to be equivalent to designing pairs of square matrices with each entry appearing equally often per row and per column.

Mutually Orthogonal Latin Squares (MOLS) were shown to provide an efficient algebraic method of systematically constructing the desired matrices and hence the trellises. Since many constructions were possible from the same initial Latin squares, two conditions based in information theory were imposed to narrow down the design. One of these conditions involved both of the trellises and provided a joint design condition. Two constructive approaches to systematically generate trellises that met these conditions were then presented. The first dealt with trellises in which the number of required states is tight. The second construction dealt with trellises that may have more states than are strictly needed. Interestingly, when the second construction is applied to trellises that have the minimum number of required states, the construction is a special case of the first one. In either case, the trellises were non-linear, non-symmetric, non-systematic and non-binary.

It was then shown that the second construction always results in trellis sections that have the group trellis section property. It was also observed that the first construction may not always yield such trellis sections. The section was closed by concluding that the group trellis section property did not guarantee good performance.

Finally, a simple extension of the Latin square based design to rates greater than 1 was considered. Since the joint trellis design condition no longer applied at these rates, each trellis was designed individually.

Chapter 4

Applications

In this chapter, four applications of the designed codes are considered: data compression, noise robust data compression, coding for the Slepian-Wolf problem and coding for the Wyner-Ziv problem. In all cases the performance is measured through simulations and the proposed methods are observed to be close to the Optimum Performance Theoretically Achievable (OPTA).

4.1 Data Compression

We first consider the problem of fixed-length to fixed-length source coding to illustrate the effectiveness of the parallel concatenated source codes. Fig. 4.1 illustrates the encoder and decoder design. A binary source with i.i.d. outputs and bias $P[X = 1] = q$ is encoded by a pair of constituent FSM encoders as given in Appendix A to form encoded sequences $\mathbf{U}_1$ and $\mathbf{U}_2$. The output of the encoders is transmitted error free to an iterative turbo decoder. There, the soft receiver assigns probabilities on the encoded data and stores these in matrices $\mathbf{r}_1$ and $\mathbf{r}_2$. Since the channel is noiseless, the matrices are almost all zero except for a single 1 per column. Iterative decoding is performed with the *a priori* probabilities p_X included in the decoding process as outlined in Section 2.8.

The performance of the scheme is measured by the achieved BER at the output of the decoder against the source entropy $H(X) = h(q) = -q\log_2 q - (1-q)\log_2(1-q)$. The BER is evaluated by simulation of at least 2000 packets of size $N = 65536$ or until 150 packet errors are received at the decoder. As a figure of reference, for an overall code rate of $2n/k$ with few packet errors, at least $131072000k$ bits were simulated and a BER of 10^{-5} would

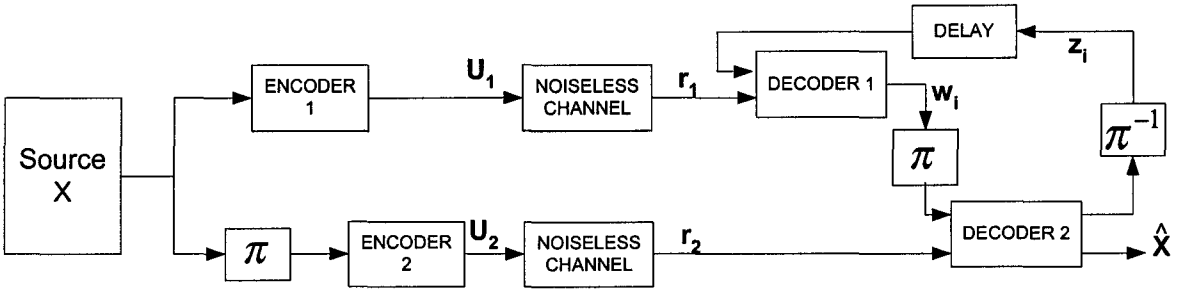

Fig. 4.1 Proposed encoder/decoder structure for fixed-length to fixed-length data compression.

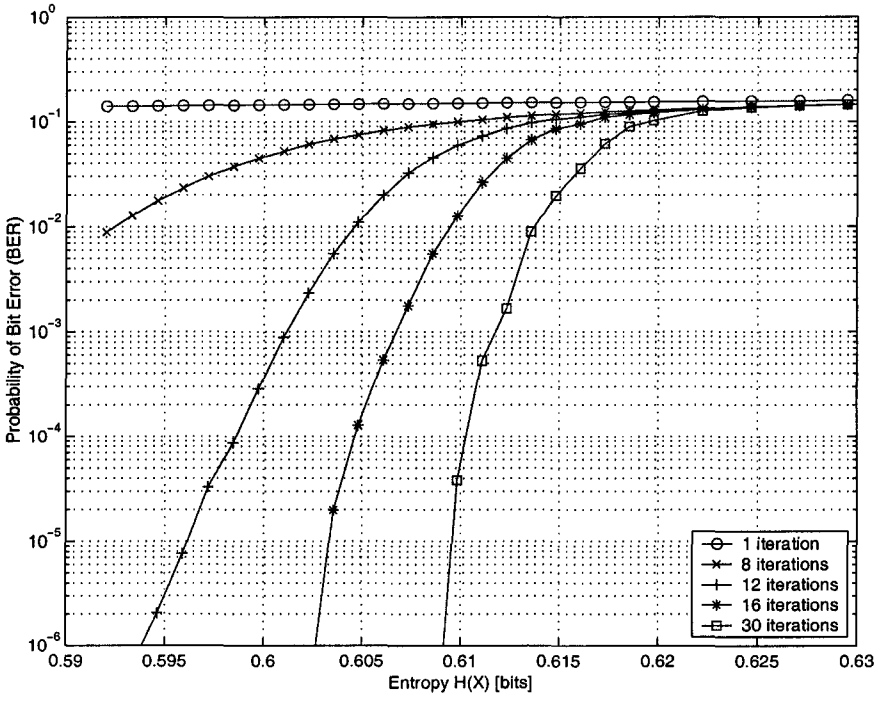

Fig. 4.2 The effect of decoding iterations on the performance of data compression with a fixed rate 2/3 code in a noiseless environment.

generate a minimum of $1310k$ bit errors. In all simulations, a spread random interleaver (see Appendix C) with spread 100 is employed unless otherwise specified.

Fig. 4.2 shows the performance achieved at the decoder for various decoding iterations for the rate 2/3 code. From the figure, it is clear that a large coding gain is achieved by using multiple iterations at the decoder: one iteration produces no noticeable difference in BER

over the illustrated range of entropies $H(X)$ while 8 iterations reduces the BER by an order of magnitude at $H(X) = 0.592$. Furthermore, 12 iterations result in a BER below 10^{-6} at the same $H(X) = 0.592$, while 16 and 30 iterations then provide a coding gain of 0.009 and 0.016 bits respectively over the performance of 12 iterations at a BER of 10^{-5}. The best result for a BER of 10^{-5} is achieved after 30 iterations at an entropy of $H(X) = 0.609$ bits. Further iterations resulted in a negligible coding gain. For a rate 2/3 code, the theoretical limit (neglecting the residual errors) is 0.6667 bits and a performance gap of 0.058 bits is observed. As a fraction of the theoretically achievable source entropy, this corresponds to a gap of 9.5%. Similar results may be observed for the rate 2/5, 2/4 and 4/5 codes from Appendix A and their performance after 10, 20 and 10 iterations respectively is illustrated in Fig. 4.3. The performance gaps vary from 0.044 bits to 0.057 bits and are listed in Table 4.1. Garcia-Frias has independently investigated the use of heavily punctured turbo codes (parallel-concatenated recursive systematic convolutional codes) [64] and the best rate reported there is 23% above the entropy of the binary memoryless source.

It is also interesting to observe how puncturing, a well known rate adjustment technique for channel codes, affects the performance of the designed codes. This is shown in Fig. 4.4 and listed in Table 4.1 for rates 0.4750 and 0.6333. These rates were obtained by puncturing (removing) 1/20 of the outputs of the rate 2/4 and 2/3 codes respectively. A few puncturing guidelines can help obtain good performance. First, it is undesirable to puncture both outputs of the same input symbol. Second, it is desirable that the punctured outputs are spread equally and are separated as far apart as possible. One can also require that the corresponding inputs of the punctured outputs are separated as far apart as possible. If too many punctured outputs are clustered together, there is little "memory" to utilize at the decoder to determine the inputs.

A simple method to achieve these guidelines is by suitably designing the interleaver. Suppose the interleaver π satisfies the following property: $n \mod k = \pi(n) \mod k$. This guarantees that puncturing an output in position l_o corresponds to an input in position l_i such that $l_o \mod k = l_i \mod k$ (regardless of which constituent encoder produced the output).

Now, if all the outputs with positions $l_o^1 \mod k = k_1$ are punctured from the output of the first constituent encoder and those in positions $l_o^2 \mod k = k_2$ from the second constituent encoder with $k_1 \neq k_2$, then no input sees its output in both streams punctured. Furthermore, the spread (or distance) between consecutive punctured outputs in

Table 4.1 Performance summary of data compression codes.

Rate	Performance ($P_e = 10^{-5}$) [bits]	OPTA Gap [bits]	$R/H(X) - 1$ [%]
2/5	0.343	0.057	16.6
0.4750	0.424	0.051	12.0
2/4	0.456	0.044	9.6
0.6333	0.571	0.062	10.9
2/3	0.609	0.058	9.5
4/5	0.744	0.056	7.5

each stream is k and the spread between the corresponding inputs is also k in each stream. However, if $k_2 = k_1 + 1$, then there are pairs of consecutive input symbols such that each one has its output punctured in a different stream. This can be resolved by requiring that $k_2 = k_1 + \lfloor k/2 \rfloor$.

The interleaver utilized for the rate 0.4750 and 0.6333 code was designed with the s-rand algorithm outlined in Appendix C and with the parameter $k = 20$ (the constraint that $n \bmod 20 = \pi(n) \bmod 20$ was added in step 5 of Appendix C). Puncturing was performed as outlined above with $k_1 = 5$ and $k_2 = 15$. From Table 4.1, one sees that the punctured codes show a slight increase in performance gap over the non-punctured equivalents, but the results are still quite close to the theoretical limit.

It is also interesting to note that the data compression scheme proposed here can achieve zero-error rate with only a minor modification. In such a situation, the encoder iteratively decodes its own output before passing the compressed data onto the decoders. This is done in order to determine the positions of the decoding errors by comparing the decoding results against the actual source message. If the positions in error are appended to the compressed data stream, the decoder can then recover the message error free. The rate increase is proportional to the BER and for a source that results in a small probability of error (i.e. when using the unmodified turbo scheme), the code rate is only marginally increased by this modification.

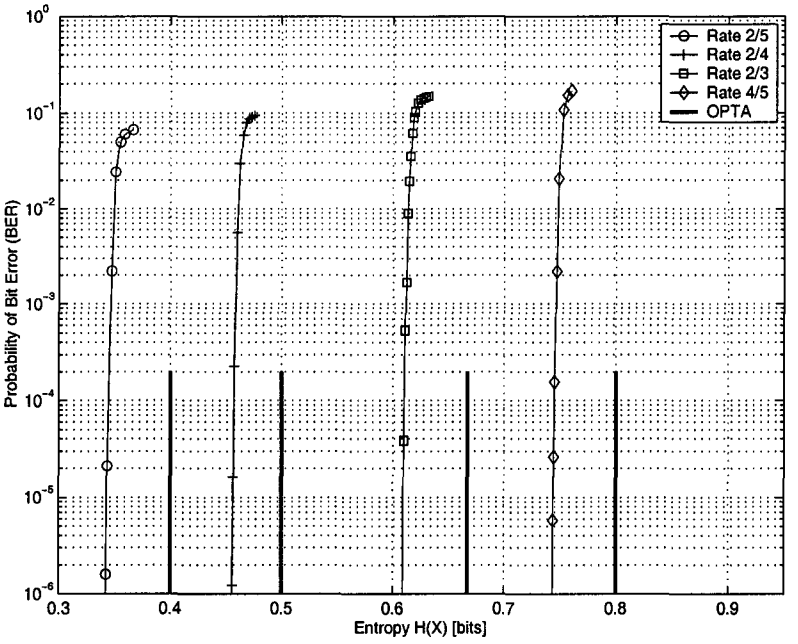

Fig. 4.3 Performance results for data compression in noiseless environment with fixed rate non-punctured codes. Performed decoding iterations are 10, 20, 30, 10 (left to right).

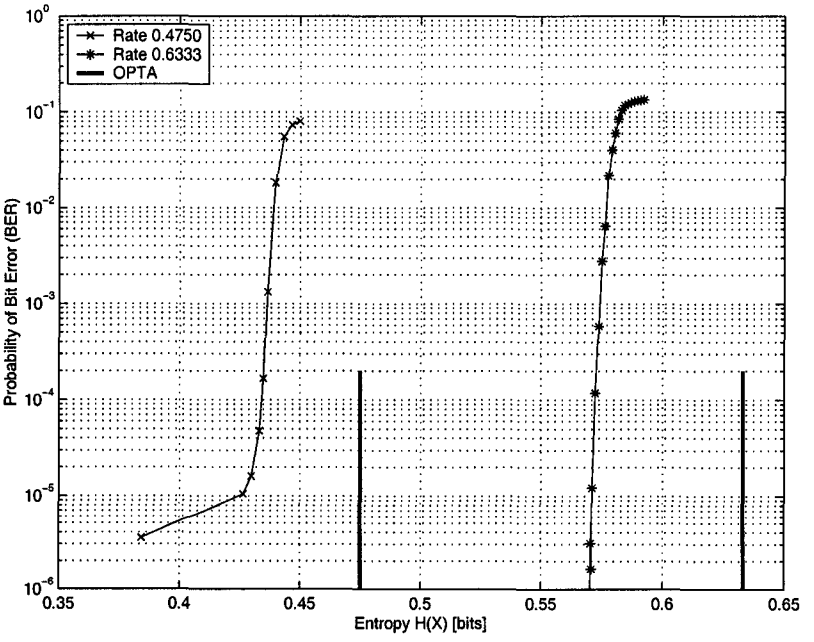

Fig. 4.4 Performance results for data compression in noiseless environment with fixed rate punctured codes after 20 decoding iterations.

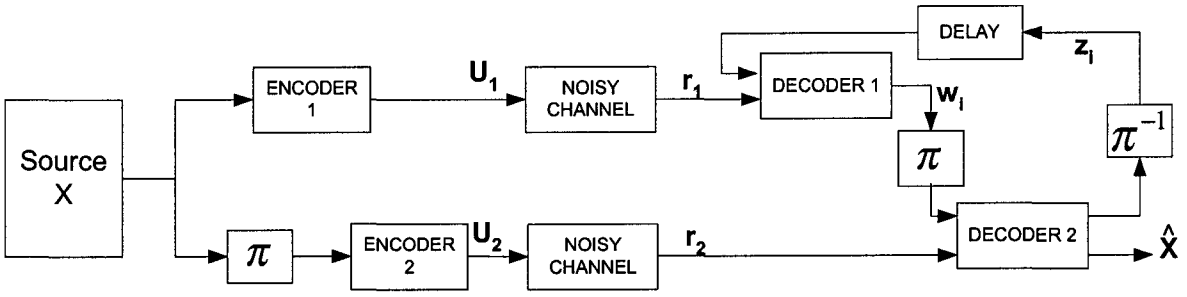

Fig. 4.5 Proposed encoder/decoder structure for noise robust data compression.

4.2 Noise Robust Data Compression

The power of turbo codes for protecting data against noise is well known. It is equally interesting to investigate how the proposed turbo source codes perform if decompression must be done from an error or noise-corrupted observation of the compressed data. This is illustrated in Fig. 4.5 and it is assumed that binary antipodal signalling is employed to transmit the compressed bit stream. In Figs. 4.6 and 4.7, the error rate is plotted as a function of E_b/N_0 in an AWGN environment where E_b is the energy per *source* bit and the variance of the noise is $N_0/2$. (The rate 0.58333 code was obtained by puncturing 1/8 of the outputs of the rate 2/3 code.)

In Fig. 4.6, as the entropy of the source is decreased by varying q from 0.11 to 0.05, it is seen that E_b/N_0 can be decreased by as much as 3.4dB and a similar result is observed for Fig. 4.7. It is clear that the *a priori* source statistics contribute a significant gain in performance. It is informative to evaluate how close to the optimal performance this noise robust compression scheme is. This can be found by numerically evaluating the capacity of an AWGN channel subject to the constraint of BPSK signalling as illustrated in Section 2.4. Tables 4.2 and 4.3 summarize the results. From these results, the difference is made as close as 1.11 dB and 1.36 dB from the OPTA respectively.

As the entropy of the source is increased to the limit that can be achieved for noiseless environments, one expects that the performance will eventually degrade somewhat. It is worthwhile noting that for the simulation of the noise robust compression scheme, an error floor occurs between the error rate of 10^{-8} and 10^{-7} with a progressive rise as the entropy of the source was increased. The fact that the probability of error cannot be made exactly zero is not surprising, as even turbo channel codes have an error floor.

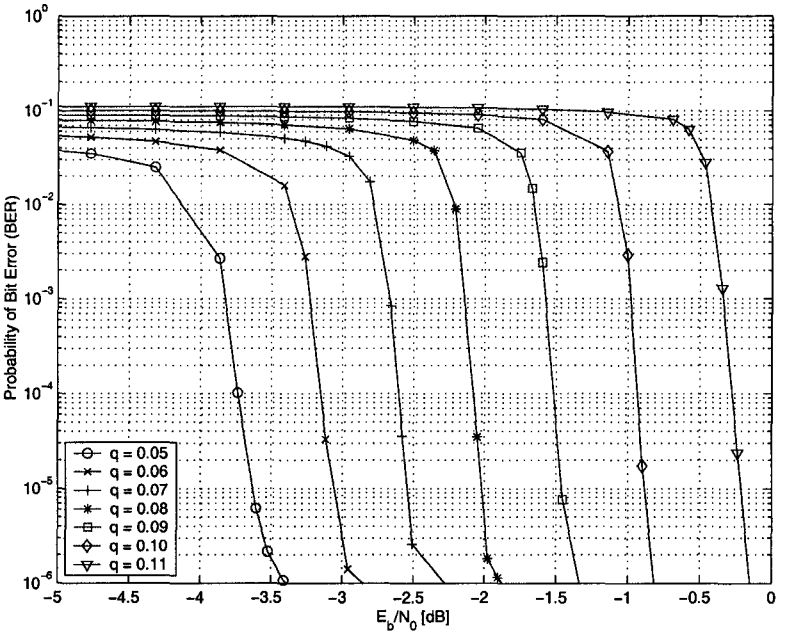

Fig. 4.6 Performance results for noise robust compression in AWGN environment with fixed rate 2/3 code. The source has a bias $P[X = 1] = q$.

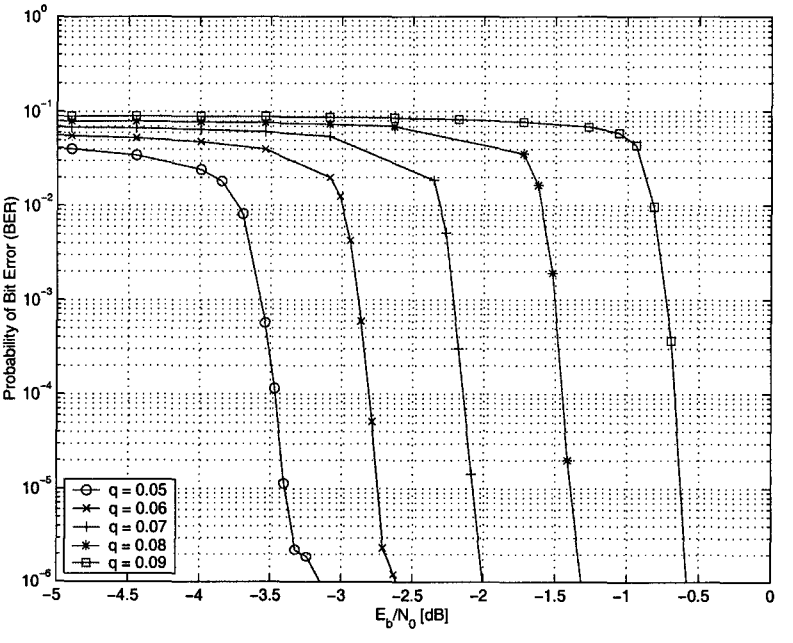

Fig. 4.7 Performance results for noise robust compression in AWGN environment with fixed rate 0.58333 code. This code was obtained by puncturing the rate 2/3 code. The source has a bias $P[X = 1] = q$.

Table 4.2 Performance summary of noise robust compression with rate 2/3 code.

q	OPTA [dB]	Performance ($P_e = 10^{-5}$) [dB]	OPTA Gap [dB]
0.05	-5.55	-3.63	1.92
0.06	-4.70	-3.06	1.64
0.07	-3.95	-2.54	1.41
0.08	-3.26	-2.02	1.24
0.09	-2.61	-1.46	1.15
0.10	-1.99	-0.88	1.11
0.11	-1.38	-0.22	1.16

Table 4.3 Performance summary of noise robust compression with rate 0.58333 code.

q	OPTA [dB]	Performance ($P_e = 10^{-5}$) [dB]	OPTA Gap [dB]
0.05	-5.28	-3.40	1.88
0.06	-4.37	-2.74	1.63
0.07	-3.54	-2.08	1.46
0.08	-2.75	-1.39	1.36
0.09	-1.99	-0.63	1.36

It is interesting to interpret the results in Tables 4.2 and 4.3. It appears that as the source bias is increased (by decreasing q), that the OPTA gap increases. One plausible explanation is that for heavily biased sources, the input is essentially always the same symbol. For long runs of inputs, the encoder cycles through the same few states, with an occasional rare change of which few states it cycles in. Since the encoder performs data compression, the output of the encoder may not allow direct observation of when there is a change of cycle states (i.e., the input that was not the most likely one produced an output identical to that of the most likely one). The performance of the code is thus degraded for heavily biased sources.

A similar observation can be made for weakly biased sources (when q is large). In those cases, the entropy of the source is such that it it is impossible to compress the source at a rate of $2n/k$ and the scheme clearly fails. Intuitively, it is clear that this failure must

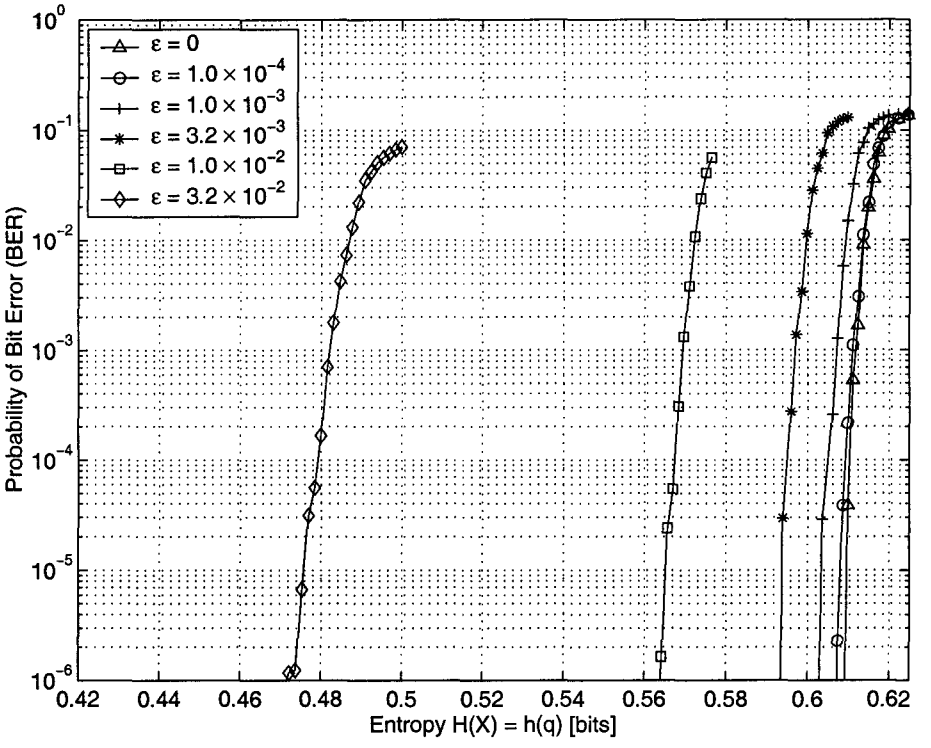

Fig. 4.8 Performance results for noise robust compression in BSC with cross-over probability ϵ and fixed rate 2/3 code.

exhibit itself by an ever increasing OPTA gap as q is increased beyond some threshold. Combining this observation with the explanation of the previous paragraph, one concludes that there is a range of q for which the scheme will perform well.

It is equally interesting to investigate the performance over binary symmetric channels (BSCs) with cross-over probability ϵ. This is illustrated in Fig. 4.8 where the BER is plotted against the entropy $H(X) = h(q)$ of the source. It is seen in Table 4.4 that the scheme performs well over a large range of cross-over probabilities, from $\epsilon = 0$ (pure data compression) to $\epsilon = 3.2 \times 10^{-2}$, with performance gaps ranging from 0.048 bits to 0.058 bits.

4.3 The Slepian-Wolf Problem

The proposed coding scheme may also be extended to the Slepian-Wolf problem, as illustrated in Fig. 4.9, where the problem is treated as a side-information coding problem [12].

Table 4.4 Performance summary of noise robust compression over BSC channel with cross-over probability ϵ for rate 2/3 code.

ϵ	OPTA [bits]	GAP [bits]
0	0.6667	0.058
1.0×10^{-4}	0.6657	0.058
1.0×10^{-3}	0.6591	0.056
3.2×10^{-2}	0.6459	0.052
1.0×10^{-2}	0.6128	0.048
3.2×10^{-1}	0.5305	0.055

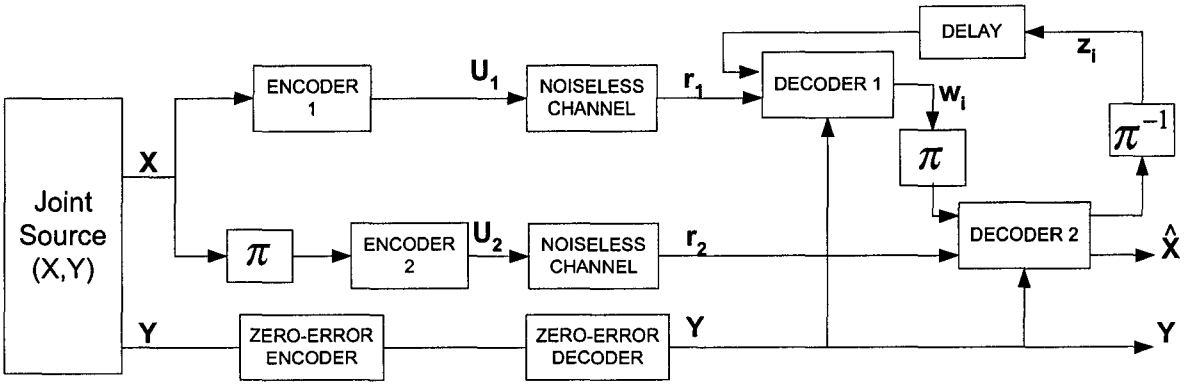

Fig. 4.9 Proposed encoder/decoder structure for Slepian-Wolf data compression.

Source X is encoded by a pair of contituent FSM encoders and source Y zero-error encoded (e.g., with the Lempel-Ziv algorithm). The receiver may then decode Y error-free. Based on the recovered Y, one may determine conditional *a posteriori* probabilities $p_{X|Y}$ which are utilized in the decoding of X in the same way as the *a priori* probabilities p_X were utilized in Section 4.1. Note that with the above coding scheme, the encoders for both X and Y have no knowledge of the source statistics as these are only employed in the decoding of X.

To test the proposed coding scheme for the Slepian-Wolf problem, it is applied, operating at a fixed rate, to different pairs of correlated binary memoryless sources (X, Y). The amount of correlation between the sources depends on their joint probability mass function

Table 4.5 Performance summary of Slepian-Wolf data compression codes.

Rate	Performance ($P_e = 10^{-5}$) [bits]	OPTA Gap [bits]	$R/H(X\|Y) - 1$ [%]
2/5	0.343	0.057	16.6
2/4	0.455	0.045	9.8
2/3	0.612	0.055	8.9
4/5	0.744	0.056	7.5

(PMF), given by the following matrix

$$P = \begin{bmatrix} 1/2 - q & q \\ q & 1/2 - q \end{bmatrix}, \tag{4.1}$$

where $q \in (0, 1/2)$ is a constant that determines the correlation between X and Y. Note, that the entropy of each separate source is 1 bit and that

$$H(X|Y) = H(Y|X) = h(2q), \tag{4.2}$$

$$H(X,Y) = 1 + h(2q), \tag{4.3}$$

where $h(.)$ is the binary entropy function. Since $H(X) = 1$, performance evaluation for this joint source (X, Y) will determine how well the scheme can utilize the side-information. For cases where $H(X) < 1$, the scheme would be a hybrid of single-source data compression and side-information at the decoder. The former was shown to perform well in Section 4.1 while the latter will now be seen to also perform well.

Fig. 4.10 shows the performance in terms of BER against the conditional entropy $H(X|Y)$ obtained by varying the parameter q for code rates of 2/5, 2/4, 2/3 and 4/5 (see Appendix A). Table 4.5 summarizes the results and a comparison with Table 4.1 shows almost identical results as for data compression. The only significant difference is for the rate 2/3 code which shows a decrease in the performance gap from 0.058 bits to 0.055 bits. In all cases, the scheme is shown to take advantage of the side-information close to the theoretical limit since without the side-information, no compression would have been possible.

A different view of what has been achieved is illustrated in Fig. 4.11, where the dotted

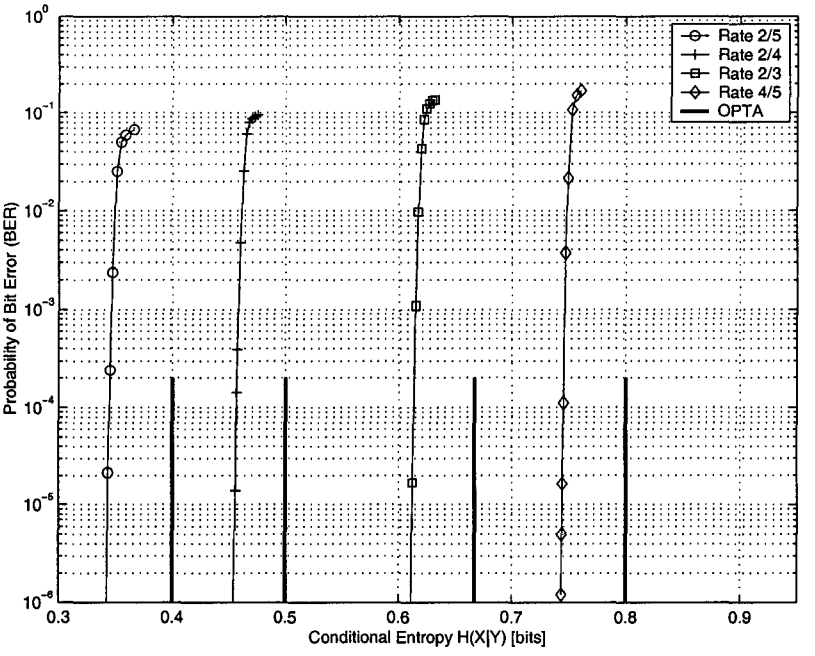

Fig. 4.10 Performance results for Slepian-Wolf compression.

line represents the reliable encoder rates that can be achieved for $H(Y|X) = H(X|Y) = 0.612$, according to the Slepian-Wolf theorem. The corners of the dashed line are the rates that have been directly achieved with BER less than 10^{-5}. The dashed line in between the corner points can be achieved by time-sharing arguments [12]. In contrast, the solid line represents the achievable region for zero-error encoding of both sources X and Y. Hence, by introducing a slight probability of error, the gain in rate is substantial.

As a figure of comparison, the rate 4/5 code presented here requires a rate in excess of $H(X|Y)$ by only 7.5%. The best result in [68] is 35.1% above $H(X|Y)$. In [63], for the same class of correlated sources, the total rate of both encoders is compared against $H(X,Y)$. There, the best result reported requires a total rate 9.8% above $H(X,Y)$ and the best result here is 3.2% above $H(X,Y)$.

Note that one could in principle make the scheme universal in the sense that neither the encoder nor the decoder requires explicit knowledge of the source statistics. If the encoders initially send a large amount of uncoded data to the decoder, the decoder may then get an accurate estimate of the statistics based on its joint observations of both X and Y. After an agreed upon amount of uncoded data has been transmitted, the encoders would then

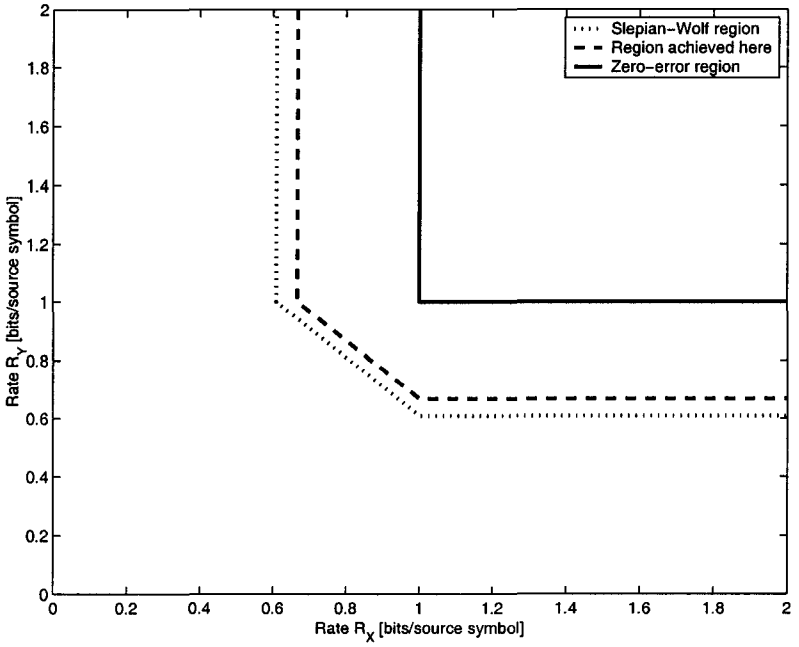

Fig. 4.11 Comparison of the Slepian-Wolf achievable region and the achieved region for the rate 2/3 code.

proceed as outlined in Fig. 4.9 and the effect of the uncoded data on the average rate would be negligible.

4.4 The Wyner-Ziv Problem

Consider a pair of independent and memoryless Gaussian sources X and U which have variances σ_X^2 and σ_U^2 respectively. Now let $Y = X + U$ and consider the coding scheme shown in Fig. 4.12, where the the continuous source X is first quantized to M levels, generating a discrete source correlated with Y. The resulting discrete source is encoded by two finite-state machine (FSM) encoders, concatenated in parallel and separated by an interleaver, and codewords $\mathbf{U}_1$ and $\mathbf{U}_2$ are transmitted over a noise-free channel. The turbo decoding principle is then used at the receiver to recover the discrete source using the coded data and side-information sequence. During the iterative decoding process, extrinsic information vectors are exchanged between decoders 1 and 2.

Let the set of M disjoint intervals $\{I_i\}_{i=1,...,M}$ partition the real line $\mathbb{R}$. The scalar quantizer f_q is a mapping $f_q : \mathbb{R} \rightarrow \{1, ..., M\}$ where $f_q(x) = i$ if $x \in I_i$. With each

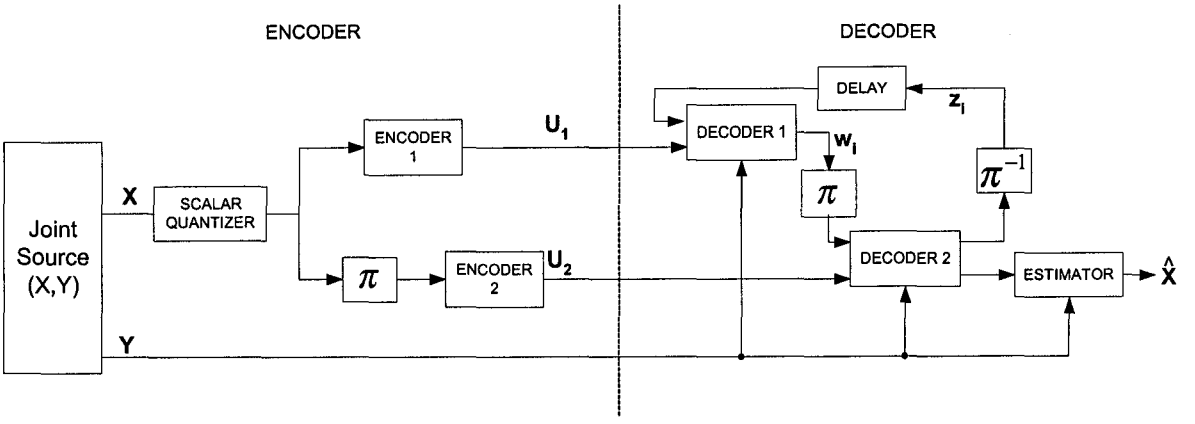

Fig. 4.12 Proposed encoder/decoder structure for Wyner-Ziv rate distortion coding with side information at the decoder.

realization of side-information y, one may evaluate the *a posteriori* conditional probability $P[x \in I_i | Y = y]$ which is employed in the iterative decoding in exactly the same way as the *a posteriori* probabilities $p_{X|Y}$ were in the Slepian-Wolf problem (see Section 4.3) and the *a priori* probabilities p_X were in data compression (see Section 4.1). Once the quantization regions have been recovered at the decoder, an estimator is utilized to determine the $\hat{X}$ which minimizes the conditional expected distortion $E[d(\hat{X}, X)|X \in I_i, Y]$. For a squared error distortion measure, this reduces to the conditional mean $\hat{x} = E[X|X \in I_i, Y = y]$.

Before proceeding to evaluate the performance of the scheme through simulation, it is interesting to determine bounds on the best performance that can be expected. Consider, for example, the rate 2/3, 2/4 and 2/5 codes which all generate 2 bits per quantized input X. As the correlation between X and Y decreases, the conditional entropy $H(f_q(X)|Y)$ increases and the scheme will clearly fail for $H(f_q(X)|Y) > 2$. Fig. 4.13 shows the conditional entropy against the correlation SNR when 8, 16 and 32 level Lloyd-Max quantization is utilized [1]. One readily observes that the conditional entropy surpasses 2 bits/sample when the correlation SNR is below 3.78 dB, 10.68 dB and 16.80 dB. By the Slepian-Wolf problem, it is impossible for the scheme to succeed at lower correlation SNRs. This implies that at best, the 8, 16 and 32 levels schemes are 3.5 dB, 2.2 dB and 2.0 dB short of the rate-distortion curve (along the correlation axis).

Under the assumption that the quantization region, $f_q(X)$, is always determined correctly from the compressed data, one can plot the rate distortion curve of this scheme with

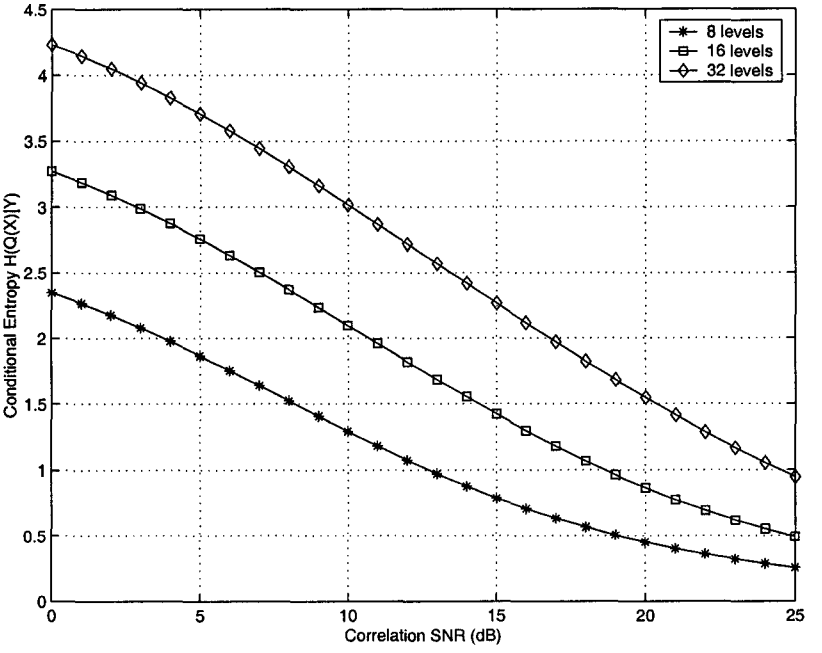

Fig. 4.13 The conditional entropy $H(f_q(X)|Y)$ for the chosen Lloyd-Max quantization levels given the side information Y as a function of the correlation SNR.

respect to that of the Wyner-Ziv bound (see Eq. (2.52)). Fig. 4.14 shows this for correlations SNRs that yield $H(f_q(X)|Y) < 2$. One may note that the decrease in distortion as the correlation SNR increases is due to the estimator. In practice, one does not expect the iterative decoder to succeed for all sources with correlation that yield $H(f_q(X)|Y) < 2$. For sources with little correlation, the side information at the decoder is insufficient to allow the iterative decoders to resolve the levels that could not be distinguished based on the encoder outputs alone (see Appendix A for the trellis).

Fig. 4.15 shows the improvement in performance as the number of decoding iterations is increased from 1 to 20 for the 8 level scheme. Increasing the number of iterations from 15 to 20 provides a coding gain of only 0.1dB in terms of correlation SNR. Most of the coding gain was obtained by the 20th iteration. Fig. 4.16 shows the simulation results after the first and last decoding iteration against the Wyner-Ziv bound for a rate $R = 2$ bits/sample. From the figure, there is a clear gain in performing multiple decoding iterations. The 8, 16 and 32 level codes achieve optimal performance at correlation SNRs of 5.2 dB, 11.8 dB and 20.1 dB. These are 1.4 dB, 1.1 dB and 3.3 dB from the scheme's best possible performance.

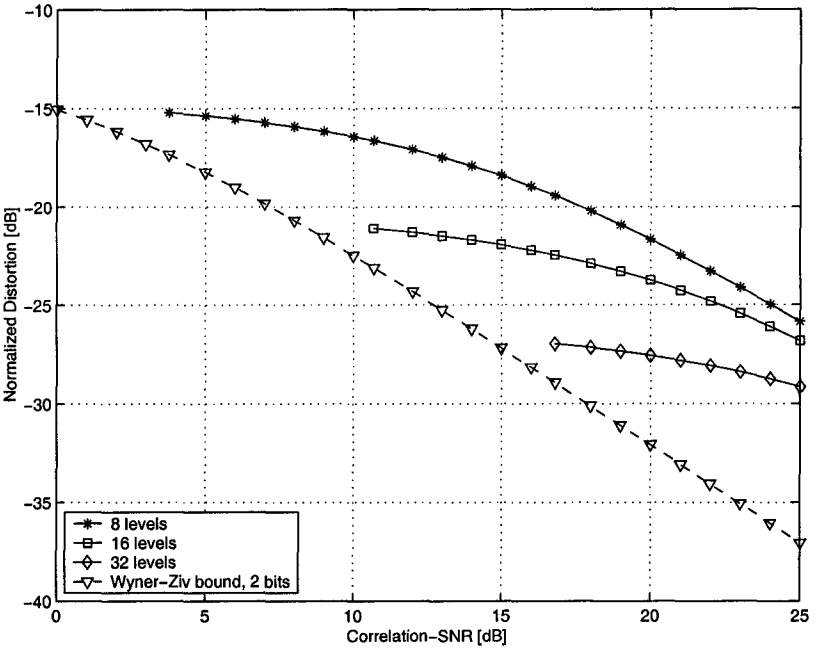

Fig. 4.14 Ideal performance achievable when the Lloyd-Max quantization levels are correctly determined for a fixed rate $R = 2$ bits/sample code.

At these points, the distortion is 3.1 dB, 2.9 dB and 4.5 dB from the optimal possible distortion of any code. Interestingly, at these operating points, the estimator is given an incorrect quantization region with probability of about 10^{-4}.

As a comparison, in the work by Servetto in [11], a distortion 1.5dB above the rate-distortion curve is reported. However, it was assumed that X and Y are strongly correlated and the results are for $\sigma_X^2/\sigma_U^2 = 100$, corresponding to a correlation SNR of 20 dB. No results have been published for lower correlation SNRs so far.

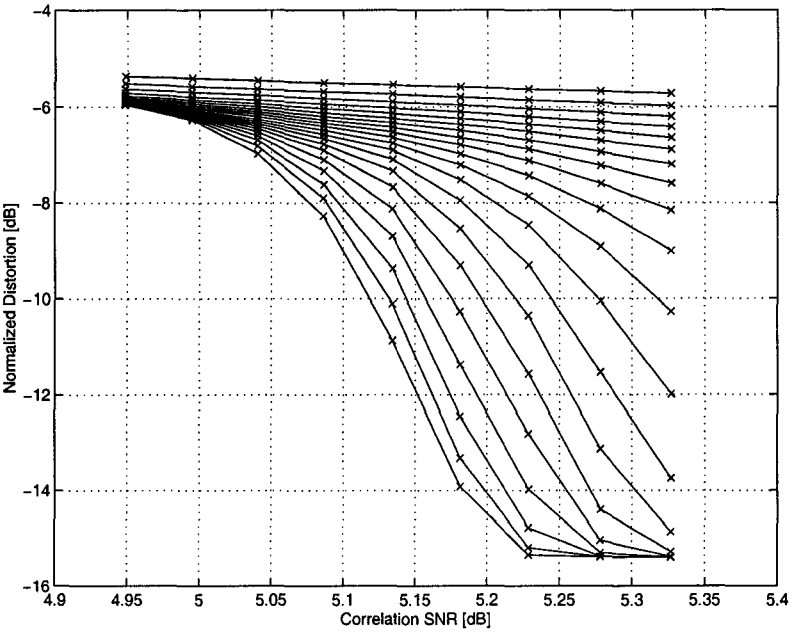

Fig. 4.15 The gain in performing up to 20 decoding iterations for the Wyner-Ziv problem with 8 level scalar quantization.

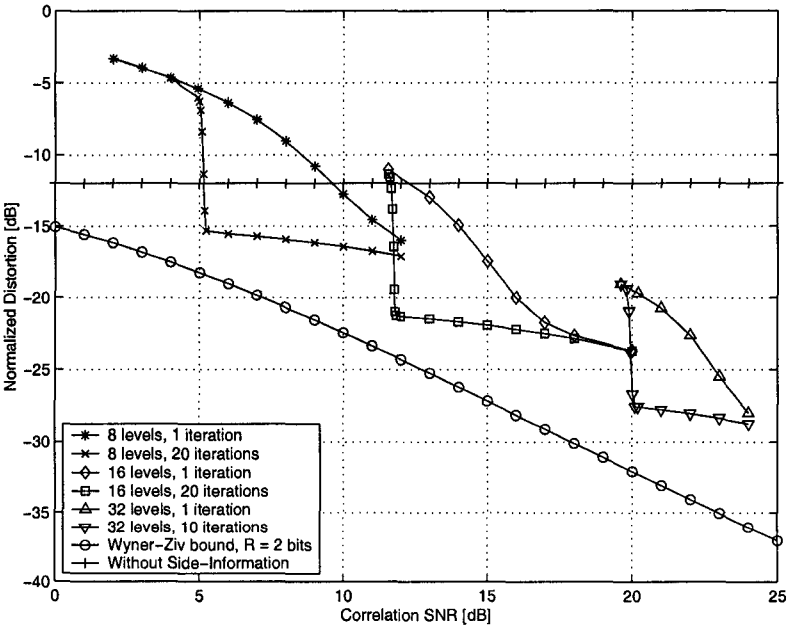

Fig. 4.16 Performance results for Wyner-Ziv coding with fixed rate $R = 2$ bits/sample codes.

Chapter 5

Conclusion

This thesis has focused on the design and applications of parallel concatenated codes for source coding, i.e. turbo source codes. As opposed to traditional source coding techniques, this method is a fixed-length to fixed-length source coding technique. The encoder does not require explicit knowledge of the source statistics and the decoder may incorporate soft information in the decoding process on the message symbols (e.g. coding for the Slepian-Wolf problem, the Wyner-Ziv problem, traditional data compression) and the coded bits (e.g. noise robust data compression). The following paragraphs will provide a summary of the work presented in this thesis and also consider future research directions.

5.1 Summary of Work

The principal contribution of this thesis has been the explicit design of parallel concatenated codes that perform data compression (i.e. codes that produce less output symbols than input symbols of the same alphabet) and their application, with a unified framework, to various coding problems in information theory. The open problems in coding theory that this thesis has treated are those of coding for the Slepian-Wolf and Wyner-Ziv problems for i.i.d. sources.

In this thesis, a design of codes that achieves near optimal Shannon limit performance for separate encoding of correlated i.i.d. binary sources has been presented. By treating the problem as a side information problem and aiming for the "corner" points of the achievable region, an array of rates is obtained through time sharing arguments. Although achieving the corner points is difficult, in this thesis, a rate in excess of $H(X|Y)$ by only 7.5% (for the

rate 4/5 code) was required for an encoder to encode X without side-information Y. The author is unaware of any published work that has achieved this performance. For example, the recently published work in [68] has a best result that is 35.1% above $H(X|Y)$.

The codes were also applied to the Wyner-Ziv problem: rate distortion of a source X when a correlated source Y not available at the encoder is available at the decoder. This was achieved by scalar quantizing X and encoding for the related Slepian-Wolf problem that ensued. The results showed excellent performance over a wide range of correlation SNRs for Gaussian sources. Results as close as 2.9 dB from the optimal possible distortion are observed. Although lattice codes have been shown to perform somewhat better than the proposed scheme for large correlation SNRs (say 20 dB [11]), the results presented here are equally good for lower correlation SNRs in the range of 5 to 12 dB for which no good performance has been published with lattice codes.

In both of the above scenarios, a discrete source is "blindly" encoded below its entropy, a rate normally required to reliably decompress the data. The reason why the decoder could recover the original message was due to the fact that additional side information, in terms of soft probabilities on the message symbols, was available at the decoder. In general, the incorporation of such statistics in a decoder is difficult. Since turbo codes are essentially a concatenation of trellis codes, they lend themselves easily to the incorporation of additional side-information of this form: each constituent decoder "biases" itself by properly incorporating the *a priori* and *a posteriori* probabilities on the message symbols.

In addition to the soft probabilities on the message symbols, it should be clear that trellis codes may also incorporate soft probabilities on the encoded symbols, e.g. as if using traditional turbo codes applied to channel coding. It follows that an interesting scenario one could consider is when both types of information are available at the decoder. This was also investigated and shown to perform well (as close as 1.1 dB from the Shannon limit). Since the scenario here is to compress the message (i.e. generate fewer outputs than inputs) yet still protect the message against channel errors, this scheme was called noise robust data compression.

5.2 Future Directions

This section will discuss future directions in which the work presented in this thesis may be extended. These include the extension to arbitrary alphabet sources, Markov sources,

improved methods for rate-distortion coding, more general network coding problems and the application of Low Density Parity Check Codes (LDPCs).

5.2.1 Arbitrary Alphabet Sources

In this work, it has always been assumed that the source alphabet had size p^k where p is a prime number. Each input symbol then produced one of p^{2n} possible outputs for a rate $2n/k$ code. Although puncturing may be used to achieve a greater flexibility in rates, it would be equally interesting if these could be achieved explicitly.

First, many interesting rates cannot be achieved with both the input and output alphabets derived from the same base p. Second, the restriction on prime powers could itself be removed. Finally, the need to puncture may eventually be eliminated.

5.2.2 Markov Sources

Throughout this thesis (and in all the published coding schemes for the Slepian-Wolf problem), it is assumed that the sources are independent and identically distributed. Although the identically distributed assumption is not very restrictive, a generalization to Markov sources (i.e. sources with memory) may be less than trivial. In principle, one could readily add the transition probabilities on the trellis in the constituent BCJR decoders. In practice, the interleaver will break up any memory and could adversely affect performance. Other methods to code for the Slepian-Wolf problem for Markov sources may be necessary.

5.2.3 Improved Methods for Rate Distortion Coding

In this thesis, rate-distortion coding was based on a two step approach. First the source was scalar quantized and then the resulting discrete output was compressed as much as possible.

One extension of this method would utilize non-contiguous quantization regions and let the side information first determine which connected subset of the quantization region is the correct one. One may observe improvements over the scheme presented in this thesis, since essentially, one could have 16 quantization levels but only 8 non-contiguous quantization regions say. However, it should first be pointed out that scalar quantization of a continuous source, followed by compression of the discrete levels cannot, in principle, achieve the rate-distortion function (as attested by figure 4.14).

The avid reader may point out that lattice codes utilize non-contiguous quantization regions and perform close to the rate-distortion function. The difference here is that lattice codes do not perform scalar quantization. The quantization regions are in R^m where m is typically 8 or more. Improved methods for rate distortion that approach the rate-distortion function will invariably be good vector quantizers.

5.2.4 General Network Coding Problems

Of all the open network coding problems, the work presented in this thesis has only been concerned with one topology. Many more encoder/decoder configurations may be investigated and most remain open even with respect to information theory (Shannon theory). Consider, for example, the problem illustrated in Fig. 5.1, where two helper functions (which are chosen by the system designer) aid in recovering Y in two separate decoders. Note that both X and Z do not need to be recovered. In [82], it was shown that the achievable region is the closure of the set of triples (R_X, R_Y, R_Z) which satisfy,

$$R_X \geq \frac{1}{n} H(U^n) \tag{5.1}$$

$$R_Y \geq \frac{1}{n} H(Y^n|U^n) \tag{5.2}$$

$$R_Y \geq \frac{1}{n} H(Y^n|V^n) \tag{5.3}$$

$$R_Z \geq \frac{1}{n} H(V^n), \tag{5.4}$$

for all positive integers n and mappings f and h.

Since, X and Z are not to be recovered at either decoder, an array of possible rates appears that was not present when treating the Slepian-Wolf problem as a side-information problem. In particular, it is not clear that time-sharing arguments can be utilized and some rates may only be achieved by (non-trivially) encoding all sources. As a simple exercise to see this, the reader is invited to fix $R_Y < H(Y)$ and investigate the achievable rates R_X and R_Z as well as the requisite functions f and g that achieve these rates.

5.2.5 Low Density Parity Check Codes

Turbo codes are no longer the only codes that have shown excellent performance with moderate complexity for channel coding. There exists a competing class of codes, known

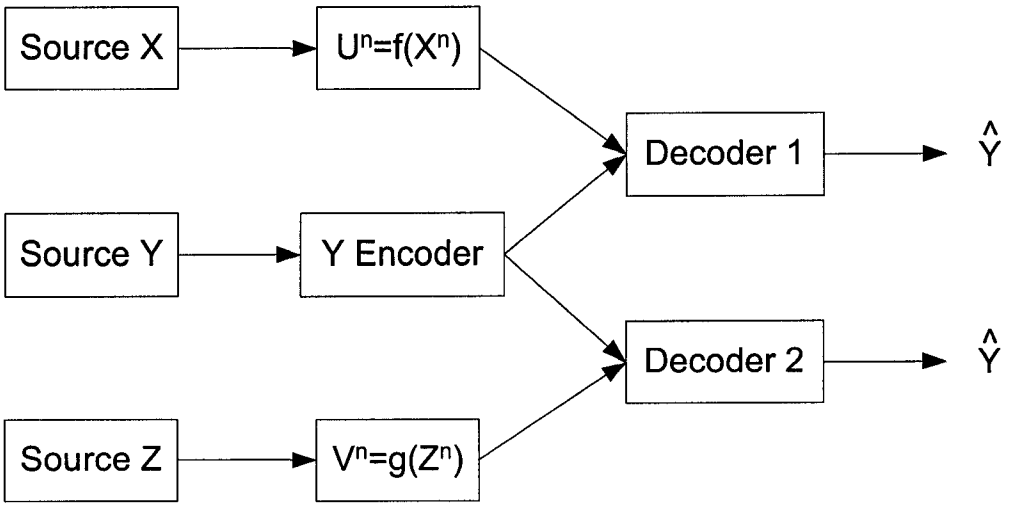

Fig. 5.1 Source network with two helpers.

as Low Density Parity Check Codes (LPDCs), that have also shown near Shannon limit performance [3].

One extension to the idea of using turbo codes for data compression would be the application of LDPCs to source coding [83]. In its simplest form, this could be accomplished by choosing parity check matrices of size $m \times n$ with $m < n$. The iterative decoder would then incorporate any soft information available at that time. Interestingly, this would almost provide a way to perform the optimal decoding of the parity check codes in the construction of [70, 71].

A more interesting problem is the application of LDPCs to rate-distortion coding. Here, the "generator matrix" G, similar to traditional LDPCs would be sparse. However, the non-zero entries would not be filled with ones but real numbers according to the distribution of the source. The encoder would iteratively search for the vector $\mathbf{x} \in \{-1, 1\}^n$ that minimizes the distortion $d(\mathbf{y}, G\mathbf{x})$.

Appendix A

Trellis

A.1 Trellises for the Designed Rate 2/3 Code

The concatenated rate 2/3 trellises are shown in Fig. A.1 and were constructed from $(a, b, c, d) = (g^3, g^0, g^2, g^6)$, where $g \in GF(p^k)$ is a primitive element of the field. The best performance was obtained by constraining the $k = 3$ length input symbols that could not be resolved directly to a Hamming distance of 2. The pairs of symbols that cannot be resolved directly are $\{0, 5\}, \{1, 2\}, \{3, 6\}$ and $\{4, 7\}$.

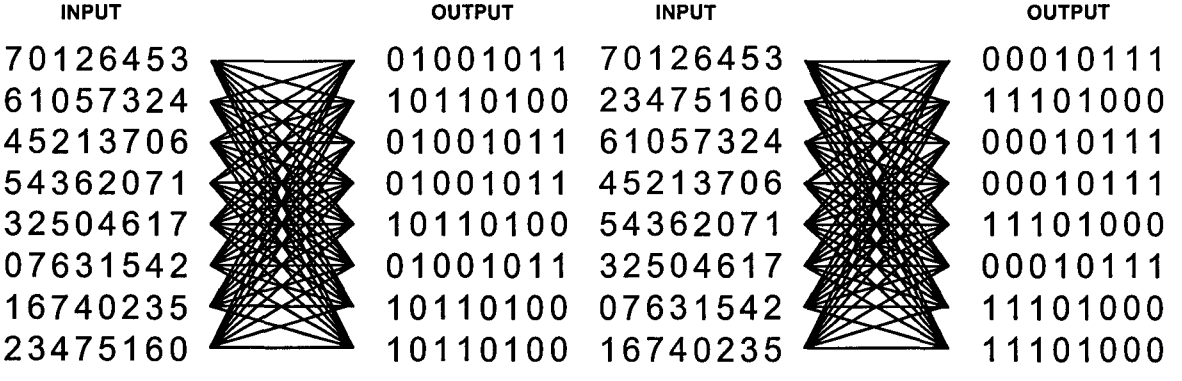

Fig. A.1 The constituent trellises for the rate 2/3 (g^3, g^0, g^2, g^6) encoder. The input edge labels are in octal notation.

A.2 Trellises for the Designed Rate 2/4 Code

The concatenated rate 2/4 trellises are shown in Fig. A.2 and were constructed from $(a, b, c, d) = (g^3, g^1, g^{10}, g^{14})$. The best performance was obtained by constraining the $k = 4$ length input symbols that could not be resolved directly to a Hamming distance of 2. The groups of input symbols that cannot be resolved directly are (in hexidecimal notation) $\{0, 6, a, c\}, \{1, 7, b, d\}, \{2, 4, 8, e\}$ and $\{3, 5, 9, f\}$.

A.3 Trellises for the Designed Rate 2/5 Code

The concatenated rate 2/5 trellises are shown in Fig. A.3 and were constructed from $(a, b, c, d) = (g^{29}, g^8, g^4, g^{17})$. Here, it is not possible that among a group of 8 symbols, all pairs have an identical Hamming distance. A minimum distance of 2 was sought. The 4 groups of input symbols that cannot be resolved directly are (in base-32 notation) $\{0, 6, a, c, i, k, o, u\}$, $\{2, 4, 8, e, g, m, q, s\}$, $\{1, 7, b, d, j, l, p, v\}$ and $\{3, 5, 9, f, h, n, r, t\}$.

A.4 Trellises for the Designed Rate 4/5 Code

The concatenated rate 4/5 trellises are shown in Fig. A.4 and were constructed from $(a, b, c, d) = (g^0, g^1, g^{21}, g^{18})$. Here, since direct observation of the outputs may narrow down the input to pairs, a minimum Hamming distance of 5 was selected. The 16 pairs of input symbols that cannot be resolved directly are (in base-32 notation) $\{0, v\}$, $\{1, u\}$, $\{2, t\}$, $\{3, s\}$, $\{4, r\}$, $\{5, q\}$, $\{6, p\}$, $\{7, o\}$, $\{8, n\}$, $\{9, m\}$, $\{a, l\}$, $\{b, k\}$, $\{c, j\}$, $\{d, i\}$, $\{e, h\}$ and $\{f, g\}$.

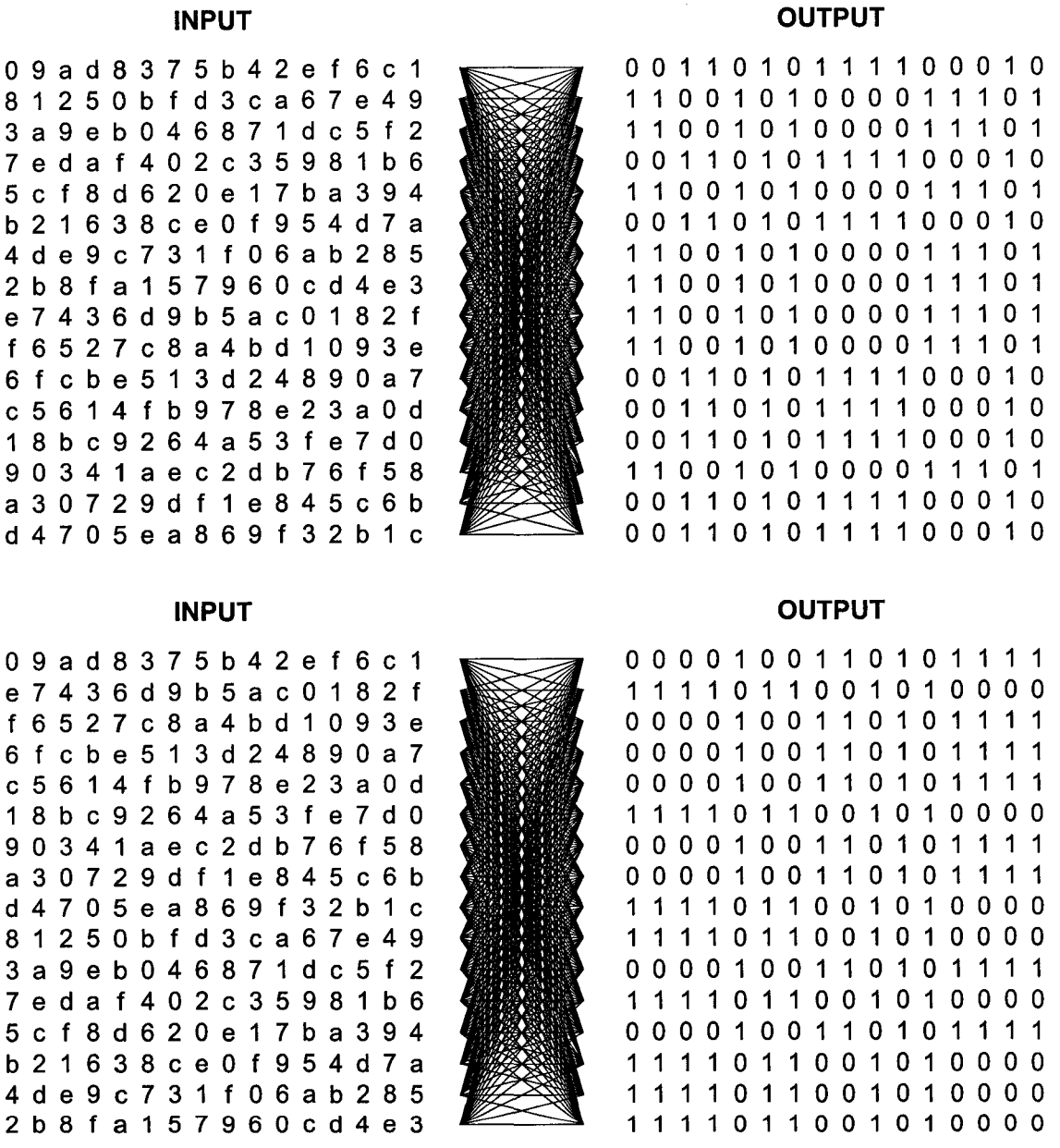

Fig. A.2 The constituent trellises for the rate 2/4 $(g^3, g^1, g^{10}, g^{14})$ encoder. The input edge labels are in hexadecimal notation.

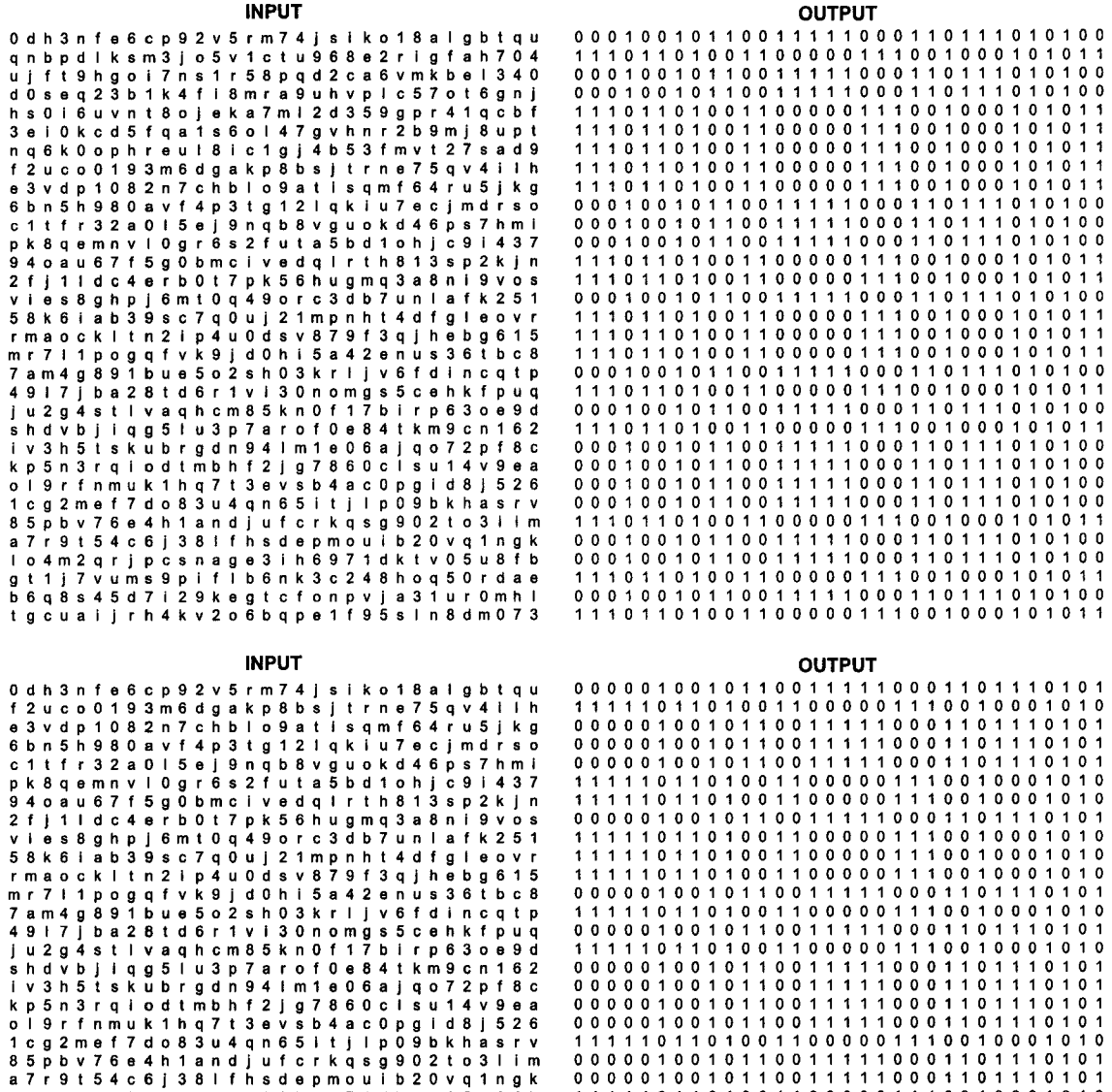

Fig. A.3 The constituent trellises for the rate 2/5 $(g^{29}, g^{8}, g^{4}, g^{17})$ encoder. The input edge labels are in base-32 notation.

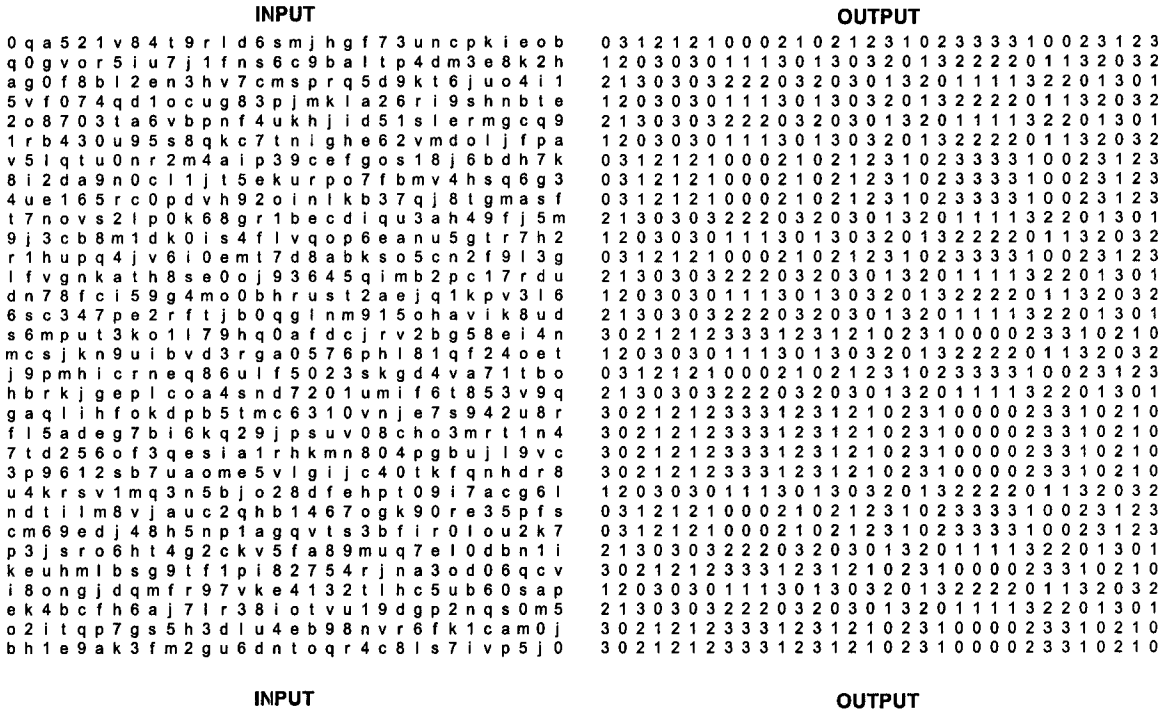

Fig. A.4 The constituent trellises for the rate 4/5 $(g^0, g^1, g^{21}, g^{18})$ encoder. The input edge labels are in base-32 notation and the output edge labels are in quaternary.

Appendix B

Important Results for Code Design

B.1 Latin Squares

Latin squares have been of mathematical interest for centuries. Perhaps the most famous problem involving Latin squares was posed by Euler in 1781: given 36 officers of six different ranks from six different regiments, can one arrange them in a square such that each row and column contain an officer of each rank and regiment? Although Euler was unable to solve the problem (though today it is known that one cannot find such an arrangement), the problem is intimately tied to Mutually Orthogonal Latin Squares (MOLS) [78, 84].

Definition B.1.1 *A Latin square of order n is an n by n matrix in which one of n symbols occurs once in each row and once in each column.*

As an example, the following are two Latin squares of order 3:

$$L_1 = \begin{matrix} a & b & c \\ c & a & b \\ b & c & a \end{matrix}, \qquad L_2 = \begin{matrix} a & b & c \\ b & c & a \\ c & a & b \end{matrix}. \tag{B.1}$$

Latin squares have found applications in many areas of science including algebra, finite geometries, coding theory, combinatorial coding design and statistics [78].

Of particular interest are a special class of pairs of Latin squares. Consider the term by term superposition of L_1 and L_2:

$$L = \begin{matrix} aa & bb & cc \\ cb & ac & ba \\ bc & ca & ab \end{matrix} . \tag{B.2}$$

A quick inspection reveals that all ordered pairs $aa, ab, ..., cc$ of symbols appear once. A pair of Latin squares that satisfies this property is said to be *mutually orthogonal.* An important theorem is discrete mathematics provides a systematic method of constructing pairs of MOLS whose order is a prime power.

Theorem B.1.1 *For $q = p^k$ a prime power, the matrices of the form $A_{i,j} = ai + j$ with $a \neq 0 \in GF(p^k)$ represent a set of $q - 1$ MOLS of order q.*

Proof: First, it is shown that if $a \neq 0$, the matrix $A_{i,j}$ represents a Latin square of order q. Suppose that there is a symbol that occurs twice in column j, i.e. at (i_1, j) and (i_2, j). Then one must have that $ai_1 + j = ai_2 + j$ and $ai_1 = ai_2$. Since $a \neq 0$, then $i_1 = i_2$. A similar argument shows that each symbol appears once per column.

Now, it is shown that $A_{i,j} = ai + j$ and $B_{i,j} = bi + j$ represent mutually orthogonal Latin squares when $a \neq b$. Suppose that both A and B are superimposed and that the same ordered pair is observed in coordinates (i_1, j_1) and (i_2, j_2). Then, one must have that,

$$ai_1 + j_1 = ai_2 + j_2 \tag{B.3}$$

$$bi_1 + j_2 = bi_2 + j_2. \tag{B.4}$$

Substraction of these equations yields $(a - b)i_1 = (a - b)i_2$ from which one concludes that $i_1 = i_2$ since $a \neq b$. If this result is combined with the above equations, one obtains that $j_1 = j_2$ and in fact, each ordered pair appears only once.

□

With the notion of MOLS at hand, it is not difficult to see that Euler's officer placement problem was to construct a pair of MOLS of order 6. Unfortunately for him, no such object exists. As a sidenote, it is easy to verify that no MOLS of order 2 exist, and Euler was also unable to construct a pair of MOLS of order 10. He conjectured that for n an odd

multiple of 2, there are no MOLS of order n. This famous conjecture was disproved in 1959 by Parker, Shrikande and Bose [78] (see page i).

A common extension of Latin squares is the notion of frequency squares. Suppose one has only m distinct symbols with $1 \leq m \leq n$. A $F(n; \lambda_1, ..., \lambda_m)$ frequency square is an $n \times n$ array in which symbol i occurs λ_i times. Obviously, $\lambda_1 + \cdots + \lambda_m = n$. The following are examples of $F(4; 2, 2)$ and $F(4; 1, 1, 2)$ frequency squares,

$$F_1 = \begin{matrix} a & b & a & b \\ b & a & b & a \\ b & b & a & a \\ a & a & b & b \end{matrix}, \qquad F_2 = \begin{matrix} a & b & c & c \\ c & c & a & b \\ b & c & c & a \\ c & a & b & c \end{matrix}. \tag{B.5}$$

B.2 A Proof of the Uniformity of States in Latin Square Based Codes with p^k States

To show that the marginal distribution of the coded symbols asymptotically approaches a uniform distribution, it was assumed that the marginal distribution of the states approached a uniform distribution as the number of stages was increased. This section will prove the latter result. Before this is accomplished, an important lemma is necessary.

Let the probability of state i in stage n be denoted by $P_n(i)$. Since all transitions are possible (since ξ is a bijection),

$$P_{n+1}(j) = \sum_{i \in GF(p^k)} p_{i,j} P_n(i), \tag{B.6}$$

where $p_{i,j}$ is the probability of transition from state i to state j. This probability depends on the actual input symbol assigned by the mapping ξ with $p_{i,j} > 0$ if no input symbol has probability 0.

Let $M(n) = \max_j P_n(j)$ and $L(n) = \min_j P_n(j)$. By Eq. (B.6), it is clear that one may produce the recursive bounds $0 \leq M(n+1) \leq M(n)$ and $L(n) \leq L(n+1) \leq 1$ from which it follows that both of the following limits exist:

$$\lim_{n \to \infty} M(n) = M \tag{B.7}$$

$$\lim_{n \to \infty} L(n) = L. \tag{B.8}$$

Lemma B.2.1 *Let $\epsilon > 0$ be given and choose N such that when $n > N$, $|M(n) - M| < \epsilon$. Then, there exists a j such that the relation $P_n(i) > M + \epsilon - 2\epsilon/p_{i,j}$ holds for all i.*

Proof: If $|M(n) - M| < \epsilon$ when $n > N$, then $M - \epsilon < M(n) < M + \epsilon$ for $n > N$. By the definition of $M(n)$ as the maximization of $P_n(j)$ over j, there must exist a j for which

$$\begin{aligned} M - \epsilon < M(n+1) &= P_{n+1}(j) && \text{(B.9)} \\ &= \sum_{i \in GF} p_{i,j} P_n(i) && \text{(B.10)} \\ &= \sum_{i \in GF, i \neq i'} p_{i,j} P_n(i) + p_{i',j} P_n(i') && \text{(B.11)} \\ &< \sum_{i \in GF, i \neq i'} p_{i,j}(M + \epsilon) + p_{i',j} P_n(i') && \text{(B.12)} \\ &= (1 - p_{i',j})(M + \epsilon) + p_{i',j} P_n(i'), && \text{(B.13)} \end{aligned}$$

where (B.13) follows since all input symbols may transit to any state (see Theorem 3.1.1a). One may then readily derive the desired result $M + \epsilon - 2\epsilon/p_{i',j} < P_n(i')$ which must hold for any i'.

□

Theorem B.2.1 *The finite state machine with p^k states given by Eqs. (3.1)-(3.4) asymptotically approaches a uniform distribution in its states.*

Proof: It will suffice to show that $L = M$. Let $\epsilon > 0$ and N be given such that $|M(n) - M| < \epsilon$ when $n > N$. By Lemma B.2.1, there exists j such that $P_n(i) > M + \epsilon - 2\epsilon/p_{i,j}$ for all i. One also has that $L(n) = \min_i P_n(i) \geq \min_i M + \epsilon - 2\epsilon/p_{i,j}$ for at least one j. As $n \to \infty$, one can make $\epsilon \to 0$ and hence $L(n) \to M$.

□

B.3 A Proof of the Uniformity of States in Latin Square Based Codes with More Than p^k States

The proof that the probability distribution of the states asymptotically approaches a uniform distribution is similar to that presented in Section B.2. In particular, if one defines

$i(j,q), q = 0, ..., p^k - 1$ to list all the p^k states that can reach state j in one transition, then the following relation is evident:

$$P_{n+1}(j) = \sum_{q=0}^{p^k-1} p_{i(j,q),j} P_n(i(j,q)) \tag{B.14}$$

Similarly, the following lemma can be shown analogously to that of Lemma B.2.1,

Lemma B.3.1 *Let $\epsilon > 0$ be given and choose N such that when $n > N$, $|M(n) - M| < \epsilon$. Then, there exists a j such that the relation $P_n(i) > M + \epsilon - 2\epsilon/p_{i(j,q),j}$ holds for all q.*

Proof: Identical to that of Lemma B.2.1.

□

The desired result may now be proven:

Theorem B.3.1 *The finite state machine with p^l states given by Eqs. (3.1)-(3.4) asymptotically approaches a uniform distribution in its states.*

Proof: Again, it is claimed that $L = M$. Let $\epsilon > 0$ be given and N such that when $n > N$, $|M(n) - M| < \epsilon$. For this to be so, by Lemma B.3.1, there must exist a j such that $P_n(i(j,q)) > M + \epsilon - 2\epsilon/p_{i(j,q),j}$ for all q. This implies that $\min_q P_n(i(j,q)) > M + \epsilon - 2\epsilon/p_{i(j,q),j}$ or that $\lim_{n\to\infty} \min_q P_n(i(j,q)) = M$. One can repeat the above argument to expand the number of states to which it applies: $\lim_{n\to\infty} \min_{q_1,q_2} P_n(i(i(j,q_1),q_2) = M$. If the Latin square is derived from a primitive element in $GF(2^l)$, since all the states of the trellis are connected, the argument can be repeated to include the probability of all the states.

□

Appendix C

Spread Interleavers

Good interleaver spreading properties are desirable for fast convergence and low-error floor performance of turbo codes [39, 85]. One interleaver design that has gained popularity is the so called "S-random" or spread interleaver [45]. It is generated with a heuristic and random algorithm, and there is no guarantee that the algorithm will be successful; several tries may be necessary.

The S-random interleaver is based on the random selection, without replacement, of integers from 0 to $N-1$. To guarantee a minimum spread S, each randomly selected integer is compared to the $S-1$ previously selected integers. If it is within $S-1$ of even one of these, it is returned to the list and a new integer is randomly chosen until the condition is satisfied [85].

The following steps produce an algorithm with complexity of $O(N)$ which, when successful, typically requires about 1 second to generate an interleaver of size $N = 65536$.

1. Construct an ordered list of numbers from 0 to $N-1$.
2. For each i from 0 to $N-1$, repeat steps 3 to 5.
3. Randomly choose a number j between i and $N-1$.
4. Exchange the numbers in positions i and j in the list.
5. Let $m = \min(i, S-1)$. Verify that the number in position i differs from the numbers in the $i-1$ to $i-m$ positions by at least S. If not, return to step 3.

References

[1] J. G. Proakis, *Digital Communications*, McGraw Hill, 4th edition, 2001.

[2] L. R. Bahl., J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 22, pp. 284–287, Mar. 1974.

[3] S.-Y. Chung, G. D. Forney Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, pp. 58–60, Feb. 2001.

[4] D. O'Shaughnessy, *Speech Communications - Human and Machine*, IEEE Press, 2nd edition, 2002.

[5] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, pp. 268–278, Mar. 1973.

[6] C. E. Shannon, "A mathematical theory of communication (Part 1)," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, July 1948.

[7] C. E. Shannon, "A mathematical theory of communication (Part 2)," *Bell Syst. Tech. J.*, vol. 27, pp. 623–656, Oct. 1948.

[8] A. K. Jain, *Fundamentals of digital image processing*, Prentice-Hall information and system sciences series, 1989.

[9] K. R. McConnell, D. Bodson, and S. Urban, *FAX: Facsimile technology and systems*, Artech House telecommunications library, 3rd edition, 1999.

[10] F. Alajaji, N. Phambo, and T. Fuja, "Channel codes that exploit the residual redundancy in CELP-encoded speech," *IEEE Trans. Speech and Audio Processing*, vol. 4, pp. 325–336, Sept. 1996.

[11] S. D. Servetto, "Quantization with side information: Lattice codes, asymptotics, and applications in wireless networks," Tech. Rep. CSL-TR-2002-1023, Cornell Computer Systems Lab, Mar. 2002.

[12] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, July 1973.

[13] A. D. Wyner, "On source coding with side-information at the decoder," *IEEE Trans. Inform. Theory*, vol. 21, pp. 294–300, May 1975.

[14] A. D. Wyner, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.

[15] R. M. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. 9, pp. 64–74, Apr. 1963.

[16] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," in *Proc. IEEE Global Telecommun. Conf.*, San Antonio, TX, Nov. 2001, pp. 1400–1404.

[17] P. Mitran and J. Bajcsy, "Design of fractional rate FSM encoders using Latin squares," in *IEEE Int. Symp. Inform. Theory — Recent Results Session*, Washington, DC, July 2001.

[18] P. Mitran and J. Bajcsy, "Turbo codes for data compression: Latin square based design and applications," In preparation for submission to *IEEE Trans. Inform. Theory*, 2002.

[19] P. Mitran and J. Bajcsy, "Near Shannon limit coding for the Slepian-Wolf problem," in *Proc. 20th Biennial Symp. on Commun.*, Kingston, Ontario, June 2002, pp. 95–99.

[20] J. Bajcsy and P. Mitran, "On Shannon source coding using parallel concatenated codes," in *IEEE Int. Symp. Inform. Theory — Recent Results Session*, Washington, DC, July 2001.

[21] P. Mitran and J. Bajcsy, "On noise robust compression using turbo-like codes," Submitted to *IEEE Commun. Lett.*, 2002.

[22] P. Mitran and J. Bajcsy, "Turbo source coding: A noise-robust approach to data compression," in *Proc. IEEE Data Compr. Conf.*, Snowbird, UT, Apr. 2002, p. 465.

[23] P. Mitran and J. Bajcsy, "Coding for the Wyner-Ziv problem with turbo-like codes," in *Proc. IEEE Int. Symp. Inform. Theory*, Lausanne, Switzerland, July 2002, p. 91.

[24] S. Verdú, "Fifty years of Shannon theory," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2057–2078, Oct. 1998.

[25] R. G. Gallager, "Claude E. Shannon: A retrospective on his life, work, and impact," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2681–2695, Nov. 2001.

[26] H. Leib, "Course notes 304-620: Information theory and coding," McGill University, 2001.

[27] T. M. Cover and J. A Thomas, *Elements of Information Theory*, Wiley series in Telecommunications, 1991.

[28] B. V. Gnedenko, *Theory of Probability*, Chelsea Publishing Company, 4th edition, 1967.

[29] R. G. Gallager, *Information Theory and Reliable Communication*, John Wiley and sons, inc., 1968.

[30] T. M. Cover, "A proof of the data compression theorem of Slepian and Wolf for ergodic sources," *IEEE Trans. Inform. Theory*, vol. 22, pp. 226–228, Mar. 1975.

[31] A. Orlitsky, "Average-case interactive communication," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1534–1547, Sept. 1992.

[32] I. Csiszár, "Linear codes for sources and source networks: Error exponents, universal coding," *IEEE Trans. Inform. Theory*, vol. 28, pp. 582–592, Jul. 1982.

[33] Y. Oohama and T. S. Han, "Universal coding for the Slepian-Wolf data compression system and the strong converse," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1908–1919, Nov. 1994.

[34] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*, Academic Press, New York, 1982.

[35] C. E. Shannon, "Coding theorems for a discrete source with fidelity criterion," *IRE Nat. Conv. Rec.*, pp. 142–163, Mar. 1959.

[36] S. Roman, *Introduction to Coding and Information Theory*, Springer-Verlag, 1996.

[37] C. Schlegel, *Trellis Coding*, Wiley-IEEE Press, 1997.

[38] P. Elias, "Coding for noisy channels," in *IRE Conv. Rec.*, 1955, pp. 37–46.

[39] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070.

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

[41] J. Boutros, G. Caire, E. Viterbo, H. Sawaya, and S. Vialle, "Turbo code at 0.03 dB from capacity limit," *Proc. IEEE Int. Symp. Inform. Theory*, p. 56, July 2002.

[42] T. Souvignier, M. Oberg, P. H. Siegel, R. E. Swanson, and J. K. Wolf, "Turbo decoding for partial response channels," *IEEE Trans. Commun.*, vol. 48, pp. 1297–1308, Aug. 2000.

[43] J. Bajcsy, C.V. Chong, D.A. Garr, J. Hunziker, and H. Kobayashi, "On iterative decoding is some existing systems," *IEEE J. Select. Areas Commun.*, pp. 883–890, May 2001.

[44] D. Divsalar and F. Pollara, "Turbo codes for deep-space communications," TDA Progress Report 42-120, Jet Propulsion Lab, Feb. 1995.

[45] D. Divsalar and F. Pollara, "Multiple turbo codes for deep-space communications," TDA Progress Report 42-121, Jet Propulsion Lab, May 1995.

[46] J. Bajcsy, "Course notes 304-623: Digital communications II," McGill University, 2001.

[47] P. Robertson, E. Villeburn, and P. Hoener, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Commun.*, Seattle, WA, June 1995, pp. 1009–1012.

[48] J. A. Erfanian, S. Pasupathy, and G. Gulak, "Reduced complexity symbol detectors with parallel structures for ISI channels," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 155–165, Feb./Mar./Apr. 1994.

[49] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, vol. 40, pp. 1098–1101, 1952.

[50] J. Ziv and A. Lempel, "A universal algorithm for data compression," *IEEE Trans. Inform. Theory*, vol. 23, no. 3, pp. 337–343, May 1977.

[51] J. Ziv and A. Lempel, "Compression of individual sequences by variable rate coding," *IEEE Trans. Inform. Theory*, vol. 24, no. 5, pp. 530–536, Sept. 1978.

[52] C. Lamy and O. Pothier, "Reduced complexity maximum a posteriori decoding of variable-length codes," in *Proc. IEEE Global Telecommun. Conf.*, San Antonio, TX, Nov. 2001, pp. 1410–1413.

[53] N. Demir and K. Sayood, "Joint source/channel coding for variable length codes," in *Proc. IEEE Data Compr. Conf.*, Snowbird, UT, Mar. 1998, pp. 139–148.

[54] M. Park and D. J. Miller, "Decoding entropy-coded symbols over noisy channels by MAP sequence estimation for asynchronous HMMs," in *Proc. Conf. on Inf. Sci. and Sys.*, Princeton, NJ, Mar. 1998, pp. 477–482.

[55] A. H. Muradm and T. E. Fuja, "Joint source-channel decoding of variable-length encoded sources," in *Proc. IEEE Inf. Theory Workshop*, Killarney, Ireland, June 1998, pp. 94–95.

[56] J. Wen and J. D. Villasenor, "Utilizing soft information in decoding of variable length codes," in *Proc. IEEE Data Compr. Conf.*, Snowbird, UT, Mar. 1999, pp. 131–139.

[57] M. Park and D. J. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," *IEEE Trans. Commun.*, vol. 48, no. 1, pp. 1–6, Jan. 2000.

[58] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," in *Proc. IEEE Data Compr. Conf.*, Snowbird, UT, Mar. 1999, pp. 158–167.

[59] S. S. Pradhan and K. Ramchandran, "Distributed source coding: Symmetric rates and applications to sensor networks," in *Proc. IEEE Data Compr. Conf.*, Snowbird, UT, Mar. 2000, pp. 363–372.

[60] A. Kh, A. Jabri, and S. Al-Issa, "Zero-error codes for correlated information sources," in *Proc. of Cryptography*, Cirencester, UK, Dec. 1997, pp. 17–22.

[61] Y.-O. Yan and T. Berger, "On instantaneous codes for zero-error coding of two correlated sources," in *Proc. IEEE Int. Symp. Inform. Theory*, Sorrento, Italy, June 2000, p. 344.

[62] P. Koulgi, E. Tuncel, S. Regunathan, and K. Rose, "Minimum redundancy zero-error source coding with side information," in *Proc. IEEE Int. Symp. Inform. Theory*, Washington, DC, June 2001, p. 282.

[63] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Commun. Lett.*, vol. 5, no. 10, pp. 417–419, Oct. 2001.

[64] J. Garcia-Frias and Y. Zhao, "Data compression of unknown single and correlated binary sources using punctured turbo codes," in *39th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, Illinois, Oct. 2001.

[65] Q. Zhao and M. Effros, "Optimal code design for lossless and near-lossless source coding in multiple access networks," in *Proc. IEEE Data Compr. Conf.*, Snowbird, UT, Mar. 2001, pp. 263–272.

[66] Q. Zhao and M. Effros, "Lossless source coding for mulptiple access networks," in *Proc. IEEE Int. Symp. Inform. Theory*, Washington, DC, June 2001, p. 285.

[67] Q. Zhao and M. Effros, "Lossless and near-lossless source coding for multiple access networks," *preprint*, 2001.

[68] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. IEEE Data Compr. Conf.*, Snowbird, UT, Apr. 2002, pp. 252–261.

[69] Y. Zhao and J. Garcia-Frias, "Data compression of correlated non-binary sources using punctured turbo codes," in *Proc. IEEE Data Compr. Conf.*, Snowbird, UT, Apr. 2002, pp. 242–251.

[70] R. Zamir and S. Shamai (Shitz), "Nested linear/lattice codes for Wyner-Ziv encoding," in *Proc. IEEE Inform. Theory Workshop*, Killarney, Ireland, June 1998, pp. 92–93.

[71] S. Shamai (Shitz), S. Verdú, and R. Zamir, "Systematic lossy source/channel coding," *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp. 564–679, Mar. 1998.

[72] S. D. Servetto, "Lattice quantization with side information," in *Proc. IEEE Data Compr. Conf.*, Snowbird, UT, Mar. 2000, pp. 510–519.

[73] X. Wang and M. T. Orchard, "Design of trellis codes for source coding with side information at the decoder," in *Proc. IEEE Data Compr. Conf.*, Snowbird, UT, Apr. 2001, pp. 361–370.

[74] R. Zamir, S. Shamai (Shitz), and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1250–1276, June 2002.

[75] S. S Pradhan and K. Ramchandran, "Group-theoretic construction and analysis of generalized coset codes for symmetric/asymmetric distributed source coding," in *Proc. Conf. on Inf. Sci. and Sys.*, Princeton, NJ, Mar. 2000.

[76] G. D. Forney and M. D. Trott, "The dynamic of group codes: State spaces, trellis diagrams and canonical encoders," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1491–1513, Sept. 1993.

[77] H. J. Ryser, *Combinatorial Mathematics*, The Carus Mathematical Monographs. The Mathematical Association of America, 1963.

[78] C. F. Laywine and G. L. Mullen, *Discrete Mathematics Using Latin Squares*, Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley and sons Inc., 1998.

[79] H.-A. Loeliger and T. Mittelholzer, "Convolutional codes over groups," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1660–1686, Nov. 1996.

[80] I. N. Herstein, *Topics in algebra*, John Wiley & Sons, 2nd edition, 1975.

[81] S. Shamai (Shitz) and S. Verdú, "The empirical distribution of good codes," *IEEE Trans. Inform. Theory*, vol. 43, no. 3, pp. 836–846, May 1997.

[82] I. Csiszár and J. Körner, "Towards a general theory of source networks," *IEEE Trans. Inform. Theory*, vol. 26, pp. 155–165, Mar. 1980.

[83] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information using low-density parity-check codes," *preprint*, 2002.

[84] N. L. Biggs, *Discrete Mathematics*, Oxford University Press, 1989.

[85] S. N. Crozier, "New high-spread high-distance interleavers for turbo-codes," in *Proc. 20th Biennial Symp. on Commun.*, Kingston, Ontario, May 2000, pp. 3–7.