ON DISTRIBUTED CODING, QUANTIZATION OF CHANNEL MEASUREMENTS AND FASTER-THAN-NYQUIST SIGNALING

A Dissertation

by

ANGELOS DIMITRIOU LIVERIS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2004

Major Subject: Electrical Engineering

ON DISTRIBUTED CODING, QUANTIZATION OF CHANNEL MEASUREMENTS AND FASTER-THAN-NYQUIST SIGNALING

A Dissertation

by

ANGELOS DIMITRIOU LIVERIS

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Costas N. Georghiades
(Chair of Committee)

Zixiang Xiong
(Member)

Krishna R. Narayanan
(Member)

Henry F. Taylor
(Member)

Jyh-Charn (Steve) Liu
(Member)

Chanan Singh
(Head of Department)

December 2004

Major Subject: Electrical Engineering

ABSTRACT

On Distributed Coding, Quantization of Channel Measurements
and Faster-than-Nyquist Signaling. (December 2004)

Angelos Dimitriou Liveris, B.S., National Technical University of Athens
Chair of Advisory Committee: Dr. Costas N. Georghiades

This dissertation considers three different aspects of modern digital communication systems and is therefore divided in three parts.

The first part is distributed coding. This part deals with source and source-channel code design issues for digital communication systems with many transmitters and one receiver or with one transmitter and one receiver but with side information at the receiver, which is not available at the transmitter. Such problems are attracting attention lately, as they constitute a way of extending the classical point-to-point communication theory to networks. In this first part of this dissertation, novel source and source-channel codes are designed by converting each of the considered distributed coding problems into an equivalent classical channel coding or classical source-channel coding problem. The proposed schemes come very close to the theoretical limits and thus, are able to exhibit some of the gains predicted by network information theory.

In the other two parts of this dissertation classical point-to-point digital communication systems are considered. The second part is quantization of coded channel measurements at the receiver. Quantization is a way to limit the accuracy of continuous-valued measurements so that they can be processed in the digital domain. Depending on the desired type of processing of the quantized data, different quantizer design criteria should be used. In this second part of this dissertation, the quantized received values from the channel are processed by the receiver, which tries to recover the transmitted information. An exhaustive comparison of several quantization criteria for this case are studied providing illuminating insight for this quantizer design problem.

The third part of this dissertation is faster-than-Nyquist signaling. The Nyquist rate in classical point-to-point bandwidth-limited digital communication systems is considered as the maximum transmission rate or signaling rate and is equal to twice the bandwidth of the channel. In this last part of the dissertation, we question this Nyquist rate limitation by transmitting at higher signaling rates through the same bandwidth. By mitigating the incurred interference due to the faster-than-Nyquist rates, gains over Nyquist rate systems are obtained.

ACKNOWLEDGMENTS

First of all, I would like to express my deepest gratitude to my advisor, Professor Costas Georghiades, because without his continuous support, guidance and patience this dissertation would never have been completed. He has helped me feel like at home in College Station and he has always been very encouraging and inspiring throughout my studies.

I am also very thankful to Professor Zixiang Xiong for introducing me to the exciting field of distributed coding. It has been a pleasure to work with him and discuss several aspects of research and life.

I would like to thank Professor Krishna Narayanan for introducing me to the field of channel coding through his graduate courses and for always being available to discuss technical and nontechnical issues.

Furthermore, I would like to express my gratitude to my other two Ph.D. committee members, Professor Henry Taylor and Professor Steve Liu, for their patience and help in several stages of my doctoral studies. I am also grateful to the other professors of the Wireless Communications group, Professor Scott Miller, Professor Erchin Serpedin, and Professor Deepa Kundur, for their help throughout my studies at Texas A&M. I would like to acknowledge the invaluable help of Sonny Matous, Tammy Carda, and Linda Currin in dealing with all the administrative procedures.

I would also like to thank Texas Instruments Inc. for funding and supporting most of the work on faster-than-Nyquist signaling and more specifically, Dr. Murtaza Ali, Dr. Alan Gatherer, and Dr. Gene Frantz.

Of course, I am also very grateful to all of my current and former colleagues, for inspiring technical and nontechnical discussions that made studying and living in College Station much more pleasant. I would particularly like to thank Hari Sankar,

Arzu Karaer, Weerakhan Tantiphaiboontana, Yongzhe Xie, Wenyan He, Nitin Nangare, Predrag Spasojević, Samuel Cheng, Vivek Gulati, Vladimir Stanković, Ching-Fu Lan, Zhixin Liu, Yong Sun, Dung Doan, Rebecca Morrison, Panayiotis Papadimitriou, Shahed Molla, Hicham Bouzekri, Murat Uysal, Yang Yang, Aravind Ganesan, Ben Lu, Jeff McDougall, Ali Shapoury, and Yan Wang.

I have been very lucky to have had the unconditional support of my parents, my brother, and my sister, as well as the rest of my friends both in Greece and in College Station, throughout my Ph.D. studies. Finally, I owe a lot to Adriana for her constant love and understanding.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

In this chapter we briefly go over the material covered in this dissertation. The dissertation is divided in three parts, as its title also suggests. The first part is distributed coding, the second is quantization of coded channel measurements and the third is faster-than-Nyquist signaling. In each of the sections that follow, more details for each part are provided and the way the material of each part has been divided into chapters is explained. This introductory chapter has been kept short and more introductory information can be found in the introduction of each of the following chapters.

A. Distributed Coding

Distributed coding in this dissertation refers to the encoding of the output of two or more physically separated sources that do not communicate with each other but send their encoded outputs to a single central point. In general, coding refers to joint source-channel coding as shown in Fig. 1, but in the special case when the channels through which the encoders communicate with the joint decoder are perfect (no distortion), we have a pure source coding (compression) problem. We will refer to this ideal source coding problem as *distributed source coding*.

There is increased interest nowadays in distributed coding, due to its possible applications in network communications, such as distributed data bases and sensor networks, and a recently introduced practical framework for distributed source coding [7]. What is remarkable about distributed coding, is that there is a great range of

The journal model is *IEEE Transactions on Automatic Control*.

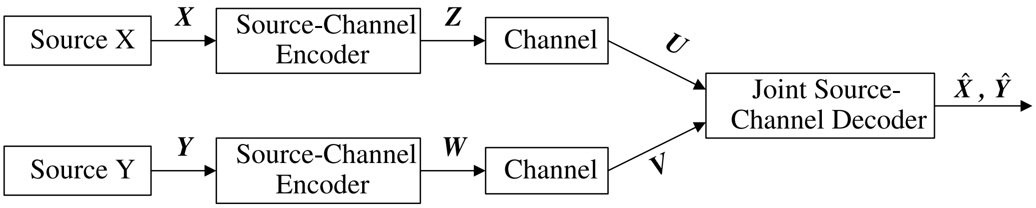

Fig. 1. Distributed coding with two correlated sources.

cases that can be considered. Depending on the number of sources, their type, the correlation statistics between their outputs, whether communication is done over perfect or noisy channels with the joint decoder, and the cost function measuring the reliability of the estimates at the output of the decoder, a unique information theoretical problem is formed for a specific combination of all these parameters. Some of these theoretical problems have not been solved yet. But even among the problems for which the theoretical limit has been given, there is a significant number for which no actual coding scheme approaching the limit has been suggested, as most of the information theoretical proofs are nonconstructive.

Suggesting such constructive schemes for some distributed coding cases is the main focus of this part of the dissertation. We will consider more extensively the case when the output of one of the sources is perfectly known at the decoder. This output can be considered as *side information* (SI) at the decoder, as shown in Fig. 2 and this problem will be refered to as *coding with side information (SI) at the decoder. Source coding with SI at the decoder* will again refer to the case of a perfect channel. Obtaining a good understanding of the ideal source coding with SI at the decoder problem helps gain more insight for the more general coding with SI at the decoder problem.

In Chapter II we assume a binary symmetric model for the sources, the side

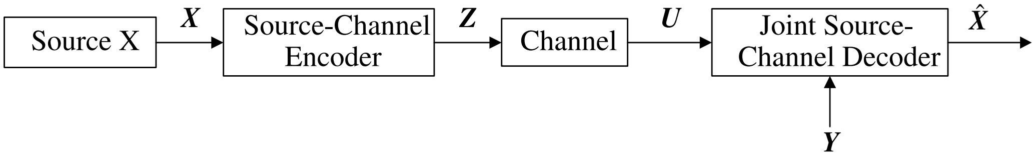

Fig. 2. Coding of a single source with side information at the decoder.

information and the correlation. This is probably the most popular statistical model considered so far for such coding problems. Several novel schemes are then proposed in Chapter II that come close to the corresponding theoretical limits, closer than competing schemes in most of the cases. The main reason for this improved performance of the suggested coding schemes is that they exploit the link that exists between most of these distributed coding problems and classical coding problems that have been extensively studied in the digital communications field.

B. Quantization of Coded Channel Measurements

To extend the distributed coding research, especially to nonbinary sources where the correlation is modeled by a discrete memoryless channel, and get more insight into coded communication systems, another part of this dissertation deals with an aspect of quantization in coded communication systems.

Quantization is the process of limiting the accuracy with which the values of a signal are described to a finite (usually small) number of bits. Quantization is used when converting an analog signal to digital, but the criteria for the quantizer design are not always the same in communication systems. They depend on what kind of processing the quantized values will experience and how the signal reconstruction, if any, will be performed. Determining the appropriate design criteria is the most

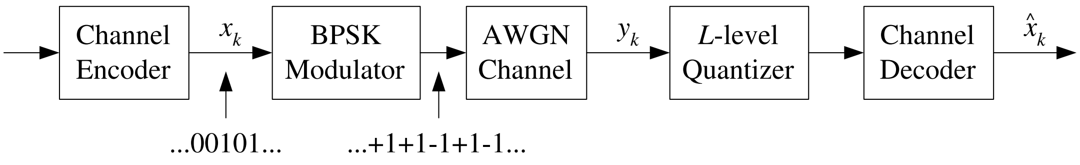

Fig. 3. Quantization of channel measurements with an L-level quantizer at the input of the channel decoder.

important part of the quantizer design and this part of the dissertation focuses on such a design for quantization of coded channel measurements.

As channel measurements in a communication system we define the analog (infinite precision) sampled received values. We assume that the system is coded, i.e., a channel code is being used, and that the channel measurements need to be quantized before being passed to the decoder, as shown in Fig. 3. Several quantizer design criteria are examined for this problem as explained in Chapter III of this dissertation, focusing on the case of the binary input additive white Gaussian noise (BIAWGN) channel, shown in Fig. 3.

The main criterion in this quantizer design problem is the error rate, i.e., the probability of making an error when trying to detect the transmitted bit sequence at the receiver. The problem with determining the bit error rate, however, is that exact analytical expressions for the bit error rate do not exist; the best one can hope for are bounds for some quite simple channel codes. Thus, one needs to resort to simulations that require days or weeks to produce results, especially for more powerful codes that require high complexity/latency decoding. Our work in this area addresses the problem of finding some quantizer design criteria that can be applied to advanced channel codes and do not require running lengthy simulations. At the same time, our approach does not sacrifice much of the optimality in the quantizer design.

Significant work has already been done in this field. The three most common general design criteria used have been:

1. the minimization of the mean-squared error (MSE) between the input and the output of the quantizer
2. the maximization of the cutoff rate of the equivalent channel between the input to the BPSK modulator and the output of the L-level quantizer
3. the maximization of the mutual information of the equivalent channel between the input to the BPSK modulator and the output of the L-level quantizer

Chapter III includes a more detailed presentation of the above three criteria and an extensive comparison between them. Such an extensive comparison has been missing from the literature. The comparison in Chapter III covers scalar quantizer designs both uniform and nonuniform, showing that the last two criteria (cutoff rate and mutual information) are almost equivalent and superior to the first one (MSE), as expected. Another important conclusion is that there is no significant difference between uniform and nonuniform scalar quantization. The last important conclusion, drawn from the extensive comparison in Chapter III for the first time ever, is the higher robustness of the second criterion (cutoff rate) over a wide range of SNRs.

C. Faster-than-Nyquist Signaling

The previous two parts of the dissertation use the discrete-time model to account for the encountered distortion due to signal transmission through the channel. When employing the discrete-time model, the equivalent binary-input discrete-time channel models the effect of the transmit filter at the transmitter, the continuous-time additive white Gaussian noise (AWGN) channel, the matched-filter at the receiver and the

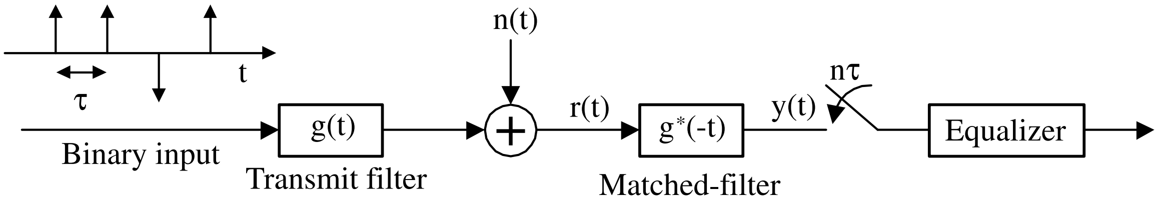

Fig. 4. Overall system for binary faster-than-Nyquist signaling.

sampler at the output of the matched-filter of Fig. 4.

In this third part of this dissertation we will use the continuous-time model of Fig. 4 and will assume bandlimited transmission. This assumption is very common in communication systems, as bandwidth along with power are the most valuable resources and therefore they should be used efficiently.

The limited bandwidth transmission constraint requires the transmit filter bandwidth to be strictly bandlimited to W Hz, i.e., the frequency response of the transmit filter should be zero for all frequencies $|f| \geq W$ (baseband transmission). In such a case it is assumed that the maximum signaling rate $\frac{1}{\tau}$ is always less than or equal to twice the transmit filter bandwidth, i.e., $\frac{1}{\tau} \leq 2W$. Equality in the last inequality is achieved for some ideal signaling pulses, such as the ideal sinc pulse which has ideal rectangular spectrum.

The maximum signaling rate $2W$ is called the *Nyquist rate*. When signaling at the Nyquist rate with ideal sinc pulses, the samples at the receiver are also taken at the same rate and there is no interference between neighboring pulses, i.e., pulses corresponding to neighboring binary inputs at the input of the transmit filter. This type of interference is called intersymbol interference (ISI).

For a more practical transmit filter, excess bandwidth needs to be used due to the more gradual roll-off of practical filters. In this case to avoid interference between neighboring pulses, the signaling rate $\frac{1}{\tau}$ has to be strictly less than the Nyquist rate, i.e., $\frac{1}{\tau} < 2W$.

In this last part of this dissertation we examine whether it is possible to reliably signal faster than the Nyquist rate. Doing so we could achieve higher rates at the expense of introducing ISI between the samples at the matched-filter output. If we could mitigate this ISI, this faster-than-Nyquist signaling will result in throughput gains that could also partly or completely be translated into coding gains. This ISI cancellation is performed by the equalizer block following the sampler in Fig. 4.

In Chapter IV we present several ways of performing equalization and joint equalization and channel decoding for faster-than-Nyquist signaling and thus for the first time ever, practical ways to exploit the faster signaling rate are proposed. All these methods exhibit the potential of faster-than-Nyquist signaling to yield gains over Nyquist rate systems and hence, its applicability as an alternative signaling scheme for increased efficiency.

CHAPTER II

DISTRIBUTED CODING

A. Introduction

Throughout this chapter we will consider the binary symmetric scenario of distributed coding which corresponds to Figs. 1 and 2 with the following simplifying assumptions:

- $\boldsymbol{X} = [X_1, X_2, \ldots, X_n]$, $\boldsymbol{Y} = [Y_1, Y_2, \ldots, Y_n]$, where the X_i's are independent and identically distributed (i.i.d.) uniform binary random variables and so are the Y_i's, i.e., Pr$[X_i = 0]$ =Pr$[X_i = 1] = \frac{1}{2}$, and similarly, Pr$[Y_i = 0]$ =Pr$[Y_i = 1] = \frac{1}{2}$.
- X_i and Y_i are correlated and their correlation is given in terms of a joint probability mass function (pmf) $p(x_i, y_i) = p(x, y) = \begin{cases} p & x \neq y \\ 1-p & x = y \end{cases}$, which will often be given in terms of the binary symmetric channel of Fig. 5.

This model has been one of the most popular in the recently published work on distributed coding. The reason is that in some cases this model appears to be practical, while in other cases, due to its simplicity, it provides significant insight into more complex distributed coding problems.

The two coding with side information (SI) at the decoder problems that this chapter focuses on are:

1. *Lossless Coding of Binary Sources*: Binary sources are sources producing binary random variables X_i whose realizations take values from the set $\mathcal{X} = \{0, 1\}$. Lossless coding means that the probability that the estimate at the output of the decoder $\hat{X}_i$ is not equal to the original output X_i of the source can be made arbitrarily small (Pr$[\hat{X}_i \neq X_i] < \epsilon$).

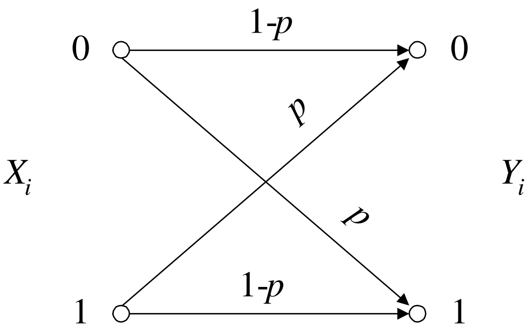

Fig. 5. Binary symmetric channel with crossover probability p modelling the correlation between the source output X_i and the second source output (or the side information) Y_i.

2. *Lossy Coding of Binary Sources*: Some limited loss d is allowed in the reconstruction $\hat{X}_i$ of X_i and this loss is measured in terms of the Hamming distance, i.e., $d = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \sum_{\hat{x}\in\hat{\mathcal{X}}} p(x,y,\hat{x})|\hat{x}-x|$, where $\mathcal{X} = \{0,1\}$, $\mathcal{Y} = \{0,1\}$ and $\hat{\mathcal{X}} = \{0,1\}$ contain all the possible values that the realizations of the random variables X_i, Y_i and $\hat{X}_i$, respectively, can take, and $p(x,y,\hat{x})$ is the joint pmf of the three binary random variables X_i, Y_i and $\hat{X}_i$.

We will also study the lossless distributed source coding problem, or Slepian-Wolf coding problem, of two and more than two binary sources. In the Slepian-Wolf coding problem of two or more than two sources the probability of reconstructing all sources' outcomes at the decoder can be made arbitrarily small.

The way lossless or lossy coding is combined with perfect or noisy channel(s) to yield a distributed compression problem or a distributed source-channel coding problem is better depicted in Table I. All the coding schemes corresponding to the entries of the Table I and presented in this chapter are novel in the sense that they were first proposed through this work and are different from existing competing approaches,

Table I. The Different Cases of Binary Symmetric Distributed Coding Considered in This Chapter

	Perfect Channel(s)	Noisy Channel(s)
Lossless Coding with SI	Asymmetric Slepian-Wolf coding (Section B, Subsection 1)	Asymmetric Slepian-Wolf coding over noisy channel (Section C)
Lossless Coding of Two Sources	Symmetric Slepian-Wolf coding (Section B, Subsection 2)	Not considered
Lossless Coding of More than Two Sources	Slepian-Wolf coding for multiple sources (Section B, Subsection 3)	Not considered
Lossy Coding with SI	Wyner-Ziv coding (Section D)	Not considered

exhibiting most of the time better performance. Subsections 1, 2 and 3 make up Section B of this chapter and then Section C and D follow. The discussion and conclusions sum up the chapter.

B. Binary Slepian-Wolf Coding

Assuming a perfect channel initially between encoder and decoder in Fig. 2, as shown in Fig. 6, the lossless source coding with SI at the decoder problem is called asymmetric Slepian-Wolf coding problem. The reason is that this problem can be viewed as a distributed source coding problem with two discrete sources by allowing time-sharing between the two sources, i.e., as the coding problem of Fig. 1 with two prefect channels. The interesting thing about this distributed source coding problem is that, under some conditions, the same overall compression can be achieved as when the two sources communicate with each other. This theoretical result is given by the Slepian-Wolf theorem [8] and this is the reason for the title of this section.

So in the first subsection we handle the asymmetric Slepian-Wolf coding problem.

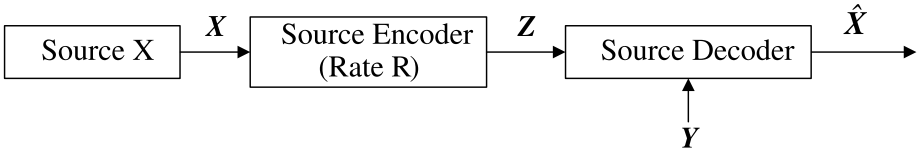

Fig. 6. System for source coding with side information at the decoder.

It mainly reduces to finding the syndromes of advanced channel codes. In Subsection 2 an alternative approach to time-sharing of asymmetric Slepian-Wolf codes for lossless distributed compression of two sources is investigated, called symmetric Slepian-Wolf coding. Subsection 3 discusses the extension of both asymmetric and symmetric Slepian-Wolf coding to more than two sources.

All the Slepian-Wolf coding approaches in this section are novel and exhibit superior performance in most of the cases to their existing competing approaches.

1. Asymmetric Coding

For the lossless source coding with SI at the decoder (asymmetric Slepian-Wolf coding) case the Slepian-Wolf limit can be written as

$$R \geq H(X_i|Y_i) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x,y) \log \frac{p(y)}{p(x,y)}, \tag{2.1}$$

where R is the allowable rate at which X_i can be compressed losslessly (in bits per source output symbol if the logarithm is base 2), $H(X_i|Y_i)$ is the conditional entropy of X_i given Y_i defined as in the last equality of (2.1) [9], $\mathcal{X}$ and $\mathcal{Y}$ contain all the possible values that the realizations of the random variables X_i and Y_i, respectively, can take, and $p(x,y)$ is the joint pmf of the two correlated random variables X_i and Y_i. This limit is the same as in the case when the SI $\boldsymbol{Y}$ in Fig. 6 is available not only at the decoder but at the encoder as well, while when there is no SI neither at the encoder nor at the decoder, the limit becomes [9]

$$R \geq H(X_i) = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)} \geq H(X_i|Y_i). \tag{2.2}$$

Thus, in general taking into account the availability of SI at the decoder when designing the source encoder and source decoder, can theoretically improve the efficiency of our system to such an extent that it is like having the side information at the encoder as well!

This interesting result of the Slepian-Wolf theorem has been known for about 30 years but it was only recently that practical approaches were proposed based on channel codes [7], i.e., a source coding problem could be solved by converting it to an equivalent channel coding problem.

Channel coding is actually employed in digital communication systems to add error protection to the transmitted signal, allowing the receiver to correct some of the errors that occur due to the distortion experienced by the signal as it propagates through the channel. Advanced channel codes are powerful codes that can achieve better protection, such as turbo codes [10] and low-density parity-check (LDPC) codes [11]. These advanced codes, invented [10] or reinvented [12, 13] in the last decade, have helped communication theory researchers come very close to the fundamental limits of communication.

To come close to the Slepian-Wolf limit, most of the researchers in this area have been trying to introduce new coding schemes based on ideas borrowed from advanced channel codes [1, 2, 3, 14, 15]. However, apart from the case of simple codes [7], there is no direct correspondence of the already proposed schemes with channel codes and so the progress already made in conventional channel coding in the attempt to approach the theoretical limits could not be exploited. Establishing this link between lossless distributed source coding and conventional channel coding is the main focus of this subsection.

Before proceeding further with this approach, we need to note that the general idea of how to achieve the Slepian-Wolf limit (2.1) is almost as old as the Slepian-Wolf theorem. Wyner [16] first suggested that the use of capacity approaching linear block codes can lead to source codes approaching the Slepian-Wolf limit, hence this concept is known as Wyner's scheme [17] (also referred to as syndrome approach [7]). The reasons that for so long no more practical schemes had been suggested, are first that nobody really clarified the idea before the elegant example in [7] was introduced and second that it was only in recent years that capacity approaching channel codes were discovered and understood.

In order to apply conventional channel codes to this source coding with SI at the decoder problem, we first need to make the overall system look like a communication system. This is done by modelling the correlation between the uncompressed source output and the side information in Fig. 6 with an equivalent channel.

In the binary symmetric setup X_i and Y_i are equiprobable binary random variables, i.e., $\Pr[X_i = 0]$ $=\Pr[X_i = 1]$ $=\Pr[Y_i = 0]$ $=\Pr[Y_i = 1] = 1/2$, and have correlation of the form $\Pr[X_i \neq Y_i] = p$. In this case the equivalent channel modelling the correlation is a binary symmetric channel (BSC) with crossover probability p, shown in Fig. 5.

If n is the codeword length and k is the information word length, a rate k/n linear binary (n, k) block code can be defined by its $(n-k) \times n$ parity check matrix $\boldsymbol{H}$, which is a matrix with binary elements. With all operations performed in the binary field, e.g., binary addition corresponds to the "XOR" operation, we define the length $(n-k)$ binary word $\boldsymbol{s}$, the *syndrome*, of an arbitrary binary length n word $\boldsymbol{w}$

as

$$\boldsymbol{s} = \boldsymbol{w}\boldsymbol{H}^T. \tag{2.3}$$

If $\boldsymbol{s} = \mathbf{0}$ (all zeros word), then $\boldsymbol{w}$ is a codeword of this code, otherwise it is not. Since the $(n-k)$ columns of $\boldsymbol{H}$ are linearly independent, there are 2^k different codewords. Each of the $2^{n-k}-1$ nonzero values of the syndrome $\boldsymbol{s}$ indexes a set of 2^k binary words and all these 2^{n-k} sets (including the zero syndrome one that contains all the codewords) are disjoint and thus, form a partition of the space of the binary words of length n. Another important property of this partition is the fact that if each set is viewed as a binary (n,k) code, it preserves the Hamming distance properties of the linear code defined by $\boldsymbol{H}$.

To employ such a code in the binary source coding with SI at the decoder setup, the source encoder can just determine the syndrome $\boldsymbol{Z}$ of the length n binary word $\boldsymbol{X}$ (source output) using the parity check matrix $\boldsymbol{H}$. Transmitting the length $(n-k)$ syndrome $\boldsymbol{Z}$ instead of the length n output $\boldsymbol{X}$, $n : (n-k)$ compression is achieved. How close this $(n-k)/n$ to the Slepian-Wolf limit $H(X_i|Y_i) = H(p) = -p\log p - (1-p)\log p$ is, has to do with how close the linear code can approach the binary symmetric channel (BSC) capacity, which is $1 - H(p)$, i.e., the rate k/n of the linear code has to be such that the decoder trying to estimate $\boldsymbol{X}$ as the most probable element of the set indexed by $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{H}^T$ given $\boldsymbol{Y}$ does not fail with probability almost equal to one.

This approach, known as Wyner's scheme, was given as a theoretical solution to the binary source coding with SI at the decoder problem [16] but only recently [7] was it used to design some practical coding schemes. However, the designs in [7] were limited to simple schemes. To exploit the recent advances in channel coding, Wyner's scheme needed to be implemented with more complicated coding schemes. This is the focus of this subsection, starting first with low-density parity-check (LDPC) codes and then considering convolutional and concatenated (turbo) codes. Both LDPC and turbo codes are state-of-the-art channel codes in coding theory nowadays.

The encoding and decoding algorithms for LDPC and concatenated codes are given next, after which, simulation results based on both schemes are presented.

a. LDPC Codes[†]

A low-density parity-check (LDPC) code is determined by its parity-check matrix $\boldsymbol{H}$ or, equivalently by its bipartite graph. An ensemble of LDPC codes is described by the degree distribution polynomials $\lambda(x)$ and $\rho(x)$ [18, 19, 20]. The bipartite graph is used in the message-passing decoding algorithm [18, 19, 20, 21]. An example of such an LDPC code graph is given in Fig. 7.

Encoding: Given $\boldsymbol{H}$, to encode, i.e., compress, an arbitrary binary input sequence, we multiply $\boldsymbol{X}$ with $\boldsymbol{H}$ [15] and find the corresponding syndrome $\boldsymbol{Z}$ (length $(n-k)$). Equivalently in the bipartite graph this can be viewed as binary addition of all the variable node values that are connected to the same check node. An example of this encoding procedure is given in Fig. 7.

Decoding: The decoder must estimate the n-length sequence $\boldsymbol{X}$ from its $(n-k)$-long syndrome $\boldsymbol{Z}$ and the corresponding n-length sequence $\boldsymbol{Y}$. We use the following notation, also shown in Fig. 8:

- x_i, $y_i \in \{0,1\}$, $i = 1, 2, \ldots, n$, are the current values of X_i and Y_i, respectively, corresponding to the i^{th} variable node v_i,

[†] ©2002 IEEE. Reprinted, with permission, from "Compression of binary sources with side information using low-density parity-check codes", Angelos D. Liveris, Zixiang Xiong, and Costas N. Georghiades, in *Proc. 2002 IEEE Global Telecommunications Conference (GLOBECOM '02)*, Nov. 2002, pp. 1300-1304.

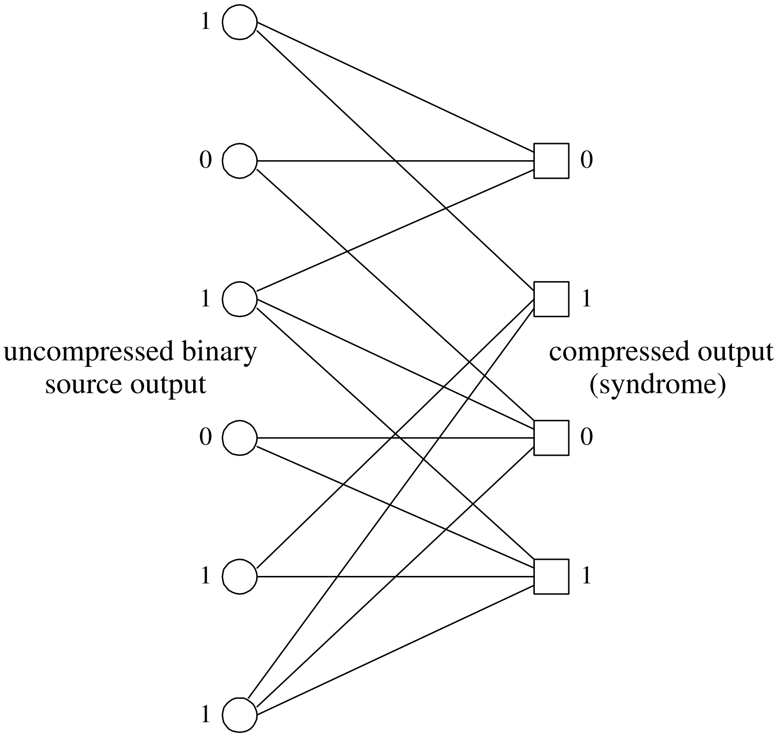
1
0
1
0
1
1
uncompressed binary source output
0
1
0
1
compressed output (syndrome)

Fig. 7. Encoding with an LDPC code using its bipartite graph: at each right (check) node all the connected left (variable) nodes are added modulo 2. Here the codeword length is only 6, $\lambda(x) = \frac{4}{7}x + \frac{3}{7}x^2$, $\rho(x) = \frac{3}{7}x^2 + \frac{4}{7}x^3$ and the rate of the original (conventional) LDPC code is 1/3, yielding a compression ratio of $3 : 2 = 1.5$.

– $l_i \in \{2, 3, \ldots\}$, $i = 1, 2, \ldots, n$, is the degree of v_i,

– $q^{\text{out}}_{i,m}$ $(q^{\text{in}}_{i,m}) \in \mathbb{R}$, $i = 1, 2, \ldots, n$, $m = 1, 2, \ldots, l_i$, is the log-likelihood ratio (LLR) sent along the m^{th} edge *from* (*to*) v_i,

– $s_j \in \{0, 1\}$, $j = 1, 2, \ldots, n-k$, is the value of Z_j corresponding to the j^{th} check node c_j, i.e., the j^{th} syndrome component,

– $r_j \in \{2, 3, \ldots\}, j = 1, 2, \ldots, n-k$, is the degree of c_j,

– $t^{\text{out}}_{j,m}$ $(t^{\text{in}}_{j,m}) \in \mathbb{R}$, $j = 1, 2, \ldots, n-k$, $m = 1, 2, \ldots, r_j$, is the LLR sent along the

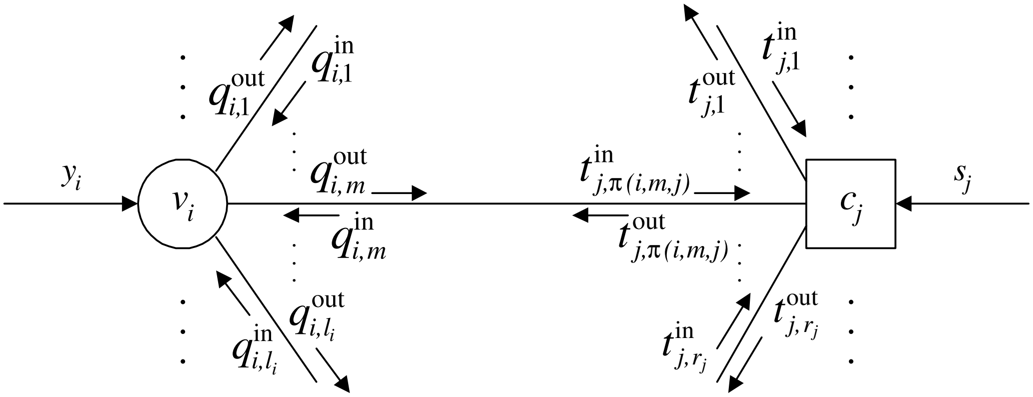

Fig. 8. Notation for the decoder operation at the i^{th} variable node v_i and j^{th} check node c_j.

m^{th} edge *from* (*to*) c_j.

Setting

$$q_{i,0} = \log \frac{\Pr[x_i = 0|y_i]}{\Pr[x_i = 1|y_i]} = (1 - 2y_i) \log \frac{1-p}{p}, \tag{2.4}$$

$i = 1, 2, \ldots, n$, $p = \Pr[x_i \neq y_i] < 0.5$, the LLR sent from the i^{th} variable node v_i along the m^{th} edge is

$$q_{i,m}^{\text{out}} = q_{i,0} + \sum_{j=1,\, j\neq m}^{l_i} q_{i,j}^{\text{in}}, \tag{2.5}$$

$m = 1, 2, \ldots, l_i$, $i = 1, 2, \ldots, n$, where initially $q_{i,j}^{\text{in}} = 0$.

The values $q_{i,m}^{\text{out}}$ are assigned to the corresponding $t_{j,\pi(i,m,j)}^{\text{in}}$ according to the connections in the bipartite graph and are then used to do the processing at the check nodes. From the "tanh rule" and the syndrome information, the LLR sent from the j^{th} check node c_j along the m^{th} edge is

$$\tanh\left(\frac{t_{j,m}^{\text{out}}}{2}\right) = (1 - 2s_j) \prod_{i=1,\, i\neq m}^{r_j} \tanh\left(\frac{t_{i,m}^{\text{in}}}{2}\right), \tag{2.6}$$

$m = 1, 2, \ldots, r_j$, $j = 1, 2, \ldots, n-k$. The inclusion of the $(1 - 2s_j)$ factor is actually the only modification to the "conventional" sum-product decoding algorithm in order to account for the syndrome information.

Now $q^{\text{in}}_{i,m} = t^{\text{out}}_{j,\pi(i,m,j)}$ for all edges in the bipartite graph, which can be used to start a new iteration and estimate x_i from

$$\hat{x}_i = \begin{cases} 0, & \text{if } q_{i,0} + \sum_{m=1}^{l_i} q^{\text{in}}_{i,m} \geq 0 \\ 1, & \text{if } q_{i,0} + \sum_{m=1}^{l_i} q^{\text{in}}_{i,m} < 0 \end{cases} . \tag{2.7}$$

b. Convolutional Codes

A general binary rate $\frac{k}{n}$ convolutional code can be given in terms of its $k \times n$ generator matrix $\boldsymbol{G}(D)$

$$\boldsymbol{G}(D) = \begin{bmatrix} g_{11}(D) & g_{12}(D) & \cdots & g_{1n}(D) \\ g_{21}(D) & g_{22}(D) & \cdots & g_{2n}(D) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ g_{k1}(D) & g_{k2}(D) & \cdots & g_{kn}(D) \end{bmatrix} \tag{2.8}$$

where each element of the table $g_{ij}(D)$ is a polynomial in D. E.g., for the rate $\frac{1}{2}$ $\boldsymbol{G}(D) = [1 + D + D^2,\ 1 + D^2]$ convolutional code $g_{11}(D) = 1 + D + D^2$ and $g_{12}(D) = 1 + D^2$.

This generator matrix can describe all possible convolutional codes, systematic and nonsystematic, punctured and nonpunctured. To account for puncturing the generator matrix has to include more than one trellis steps and so the generator matrix might become quite large. To avoid confusion in this part on convolutional codes and the next one one on concatenated convolutional codes, we denote with capital letters N and K the convolutional codeword length and the corresponding information sequence length, respectively, while small $n < N$ and $k < K$ are used to denote the number of convolutional coded bits and the corresponding number of information bits, respectively, in one trellis step or a few more if the convolutional code is punctured. In both cases the convolutional code rate is given as $\frac{k}{n} = \frac{K}{N}$.

In channel coding, channel encoding is done by taking the information sequence $\boldsymbol{a} = [a_0, a_1, \ldots, a_{K-1}]$ and passing it through the convolutional encoder to produce the coded bit sequence. Assuming that the length of the input bit sequence K is a multiple of k, i.e., $K = k\,m$, where m is a positive integer, the resulting $N = n\,m$ coded bits from the rate $\frac{k}{n}$ convolutional code can be determined from

$$[x_0(D),\, x_1(D), \ldots, x_{n-1}(D)] = [a_0(D),\, a_1(D), \ldots, a_{k-1}(D)]\; \boldsymbol{G}(D)\,, \tag{2.9}$$

where $x_i(D) = x_i + x_{i+n}\,D + x_{i+2n}\,D^2 + \ldots + x_{i+n(m-1)}\,D^{m-1}$ with $\boldsymbol{x} = [x_0, x_1, \ldots, x_{N-1}]$ being the coded bit sequence and similarly, $a_i(D) = \sum_{j=0}^{m-1} a_{i+kj}\,D^j$.

Without loss of generality, we can assume that the first k columns of the generator matrix $\boldsymbol{G}(D)$ are linearly independent. If they are not, it is possible to rearrange the columns in $\boldsymbol{G}(D)$ so that the first k are linearly independent. We will denote this invertible $k \times k$ submatrix of $\boldsymbol{G}(D)$ as $\boldsymbol{G}_1(D)$ and the remaining $k \times (n-k)$ submatrix of $\boldsymbol{G}(D)$ as $\boldsymbol{G}_2(D)$, i.e.,

$$\boldsymbol{G}(D) = [\boldsymbol{G}_1(D) \quad \boldsymbol{G}_2(D)]\ . \tag{2.10}$$

Then one possible choice for the $(n-k) \times n$ parity-check matrix $\boldsymbol{H}(D)$ of the rate $\frac{k}{n}$ convolutional code $\boldsymbol{G}(D)$ is

$$\boldsymbol{H}(D) = \Big[\boldsymbol{G}_2^T(D)\ \big[\boldsymbol{G}_1^T(D)\big]^{-1} \qquad \boldsymbol{I}_{n-k}\Big]\ , \tag{2.11}$$

where $\boldsymbol{I}_{n-k}$ is the $(n-k) \times (n-k)$ identity matrix and the superscript ^{T} denotes matrix transpose. It is easy to confirm that $\boldsymbol{H}(D)\,\boldsymbol{G}^T(D) = \mathbf{0}_{(n-k),k}$, where $\mathbf{0}_{(n-k),k}$

is the $(n-k) \times k$ all-zeros matrix.

The inversion of $\boldsymbol{G}_1(D)$ as well as the possible column rearrangement in $\boldsymbol{G}(D)$ to form an invertible $\boldsymbol{G}_1(D)$ should not be difficult to perform when $\boldsymbol{G}(D)$ is a small matrix. We will see next how the difficulty in inverting a large matrix $\boldsymbol{G}(D)$, which might come up in a randomly punctured convolutional code, can be overcome by using the trellis of $\boldsymbol{G}(D)$. We assume that when the convolutional code is systematic $\boldsymbol{G}_1(D) = \boldsymbol{I}_k$.

For example, for the rate $\frac{1}{2}$ convolutional code $[1+D+D^2,\ 1+D^2]$, we would get from (2.11)

$$\boldsymbol{H}(D) = \left[\frac{1+D^2}{1+D+D^2}\,, \quad 1\right] \tag{2.12}$$

Encoding: Using the above matrix formulation and following Wyner's scheme the syndromes of a binary length $N = n\,m$ sequence $\boldsymbol{x} = [x_0, x_1, \ldots, x_{N-1}]$ with respect to the rate $\frac{k}{n}$ convolutional code $\boldsymbol{G}(D)$ can be determined as follows. The first step is to form n different length m subsequences $\boldsymbol{x}_i = [x_i, x_{i+n}, \ldots, x_{i+n(m-1)}]$, $i = 0, 1, \ldots, n-1$. Each of these subsequences can be written in polynomial form as

$$x_i(D) = x_i + x_{i+n}\,D + x_{i+2n}\,D^2 + \ldots + x_{i+n(m-1)}\,D^{m-1}\,. \tag{2.13}$$

Then the asymmetric Slepian-Wolf encoder, which is just a syndrome former according to Wyner's scheme, is performed as [22]

$$[s_0(D),\, s_1(D), \ldots, s_{n-k-1}(D)] = [x_0(D),\, x_1(D), \ldots, x_{n-1}(D)]\ \boldsymbol{H}^T(D)\,, \tag{2.14}$$

where

$$s_i(D) = s_i + s_{i+n-k}\,D + s_{i+2(n-k)}\,D^2 + \ldots + s_{i+(n-k)(m-1)}\,D^{m-1} \tag{2.15}$$

with $i = 0, 1, \ldots, n-k-1$, and $\boldsymbol{s} = [s_0, s_1, \ldots, s_{N-K-1}]$, being the length $(N-K)$

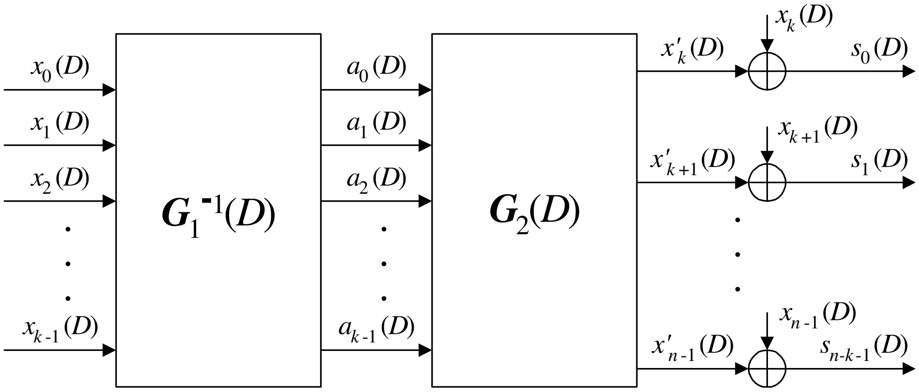

Fig. 9. Forming the syndromes of the rate $\frac{k}{n}$ convolutional code $\boldsymbol{G}(D) = [\boldsymbol{G}_1(D) \quad \boldsymbol{G}_2(D)]$ using equation (2.16).

sequence of the resulting syndrome bits.

Based on all the above, we can rewrite equation (2.14) as

$$\begin{aligned}[s_0(D),\, s_1(D), \ldots, s_{n-k-1}(D)] \;&=\; [x_0(D),\, x_1(D), \ldots, x_{k-1}(D)]\; \boldsymbol{G}_1^{-1}(D)\, \boldsymbol{G}_2(D) \\ &\quad + [x_k(D),\, x_{k+1}(D), \ldots, x_{n-1}(D)]\;. \qquad (2.16)\end{aligned}$$

The importance of the last equation in forming the syndromes is the following: for every n bits from the sequence $\boldsymbol{x}$ the first k can be used to determine the first term of the right side of the last equation using the rate $\frac{n-k}{k}$ nonsystematic convolutional code $\boldsymbol{G}_1^{-1}(D)\, \boldsymbol{G}_2(D)$.[1] The $(n-k)$ resulting bits from this convolutional code encoding can then be added to the remaining $(n-k)$ bits of these n bits from the sequence $\boldsymbol{x}$ to form the corresponding $(n-k)$ syndrome bits. The way this syndrome former works is depicted in Fig. 9.

The importance of this method of forming the syndromes is that the correspond-

[1] The rate of this convolutional code does not have to be less than one.

ing information sequence $\boldsymbol{a}$ is first formed through $\boldsymbol{G}_1^{-1}(D)$, as shown in Fig. 9, i.e.,

$$[x_0(D),\, x_1(D), \ldots, x_{k-1}(D)] = [a_0(D),\, a_1(D), \ldots, a_{k-1}(D)]\ \ \boldsymbol{G}_1(D)\,. \tag{2.17}$$

This property will be very helpful when two convolutional codes are concatenated, as discussed in the next part on concatenated codes, where the information sequences are interleaved. Furthermore, this one-to-one correspondence between the $k\,m$ long information sequence $\boldsymbol{a}$ and $k\,m$ bits of the sequence $\boldsymbol{x}$ means that since the information sequence $\boldsymbol{a}$ determines the path through the convolutional code trellis, the corresponding $k\,m$ bits of the sequence $\boldsymbol{x}$ can equivalently play this role.

More specifically, the first k bits from the sequence $\boldsymbol{x}$, i.e., the k bits corresponding to the coefficients of D^0 in the polynomials $x_0(D),\, x_1(D), \ldots, x_{k-1}(D)$, can be mapped to a unique combination of the first k bits of the information sequence $\boldsymbol{a}$. Similarly, the k bits corresponding to the coefficients of D^1 in the polynomials $x_0(D),\, x_1(D), \ldots, x_{k-1}(D)$, can be mapped to a unique combination of the second set of k bits in the information sequence $\boldsymbol{a}$, and so on. This means that the trellis transitions in the linear convolutional code trellis can be selected from the properly chosen bits of the sequence $\boldsymbol{x}$ and thus, these bits can select the path through the linear convolutional code trellis. E.g., in Fig. 10 it is shown how the even positioned bits of $\boldsymbol{x}$, i.e., $x_0, x_2, x_4, \ldots, x_{N-2}$, can select the path through the trellis of the linear convolutional code $[1 + D + D^2,\ 1 + D^2]$. In other words, a subsequence of $\boldsymbol{x}$ can uniquely identify the path through the trellis that should be followed and thus, also the information sequence $\boldsymbol{a}$ corresponding to this path.

Then, based on the selected path or equivalently, this information sequence $\boldsymbol{a}$, the remaining part of the codeword of the linear convolutional code $[\boldsymbol{G}_1(D)\ \ \boldsymbol{G}_2(D)]$

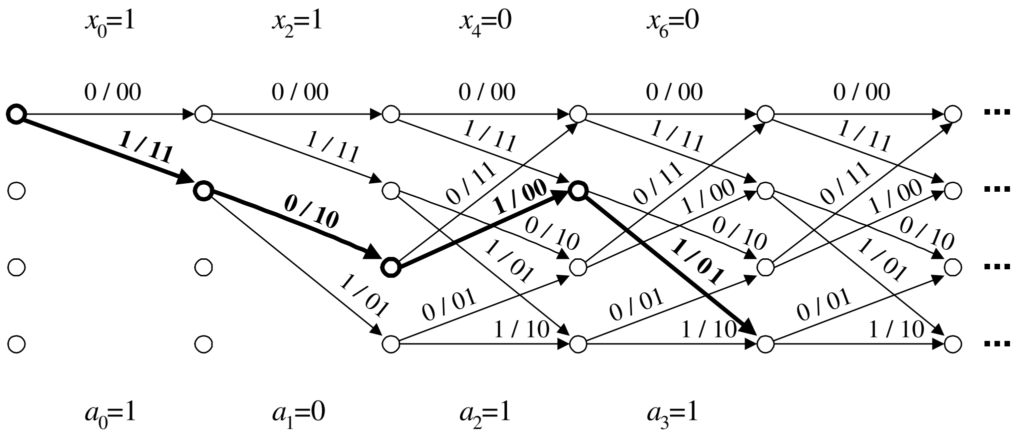

Fig. 10. The bits x_0, x_2, x_4, x_6 are used to determine the bold path through the encoding trellis of the linear convolutional code $[1 + D + D^2,\ 1 + D^2]$, as they are equivalent to a_0, a_1, a_2, a_3. The numbers over the transitions indicate the single input (information) bit and the two output (coded) bits when the trellis is used for channel encoding, i.e., they are in the form $a_i/x_{2i}x'_{2i+1}$.

corresponding to $\boldsymbol{a}$ is

$$\left[x'_k(D),\, x'_{k+1}(D), \ldots, x'_{n-1}(D)\right] = [a_0(D),\, a_1(D), \ldots, a_{k-1}(D)]\ \boldsymbol{G}_2(D)\,. \qquad (2.18)$$

The combination of the bits from $x_0(D),\ x_1(D), \ldots, x_{k-1}(D)$ and $x'_k(D),\ x'_{k+1}(D), \ldots,$ $x'_{n-1}(D)$ would yield a valid codeword of the linear convolutional code $[\boldsymbol{G}_1(D)\ \ \boldsymbol{G}_2(D)]$, as

$$\left[x_0(D),\, x_1(D), \ldots, x_{k-1}(D),\, x'_k(D),\, x'_{k+1}(D), \ldots, x'_{n-1}(D)\right] = \\ = [a_0(D),\, a_1(D), \ldots, a_{k-1}(D)] \cdot [\boldsymbol{G}_1(D)\ \ \boldsymbol{G}_2(D)]\ . \qquad (2.19)$$

In Fig. 10 the bits in $x'_k(D),\ x'_{k+1}(D), \ldots, x'_{n-1}(D)$ correspond to the second coded bit, i.e., the rightmost bit over each trellis transition.

The syndromes of the sequence $\boldsymbol{x}$ can be determined from the difference of the

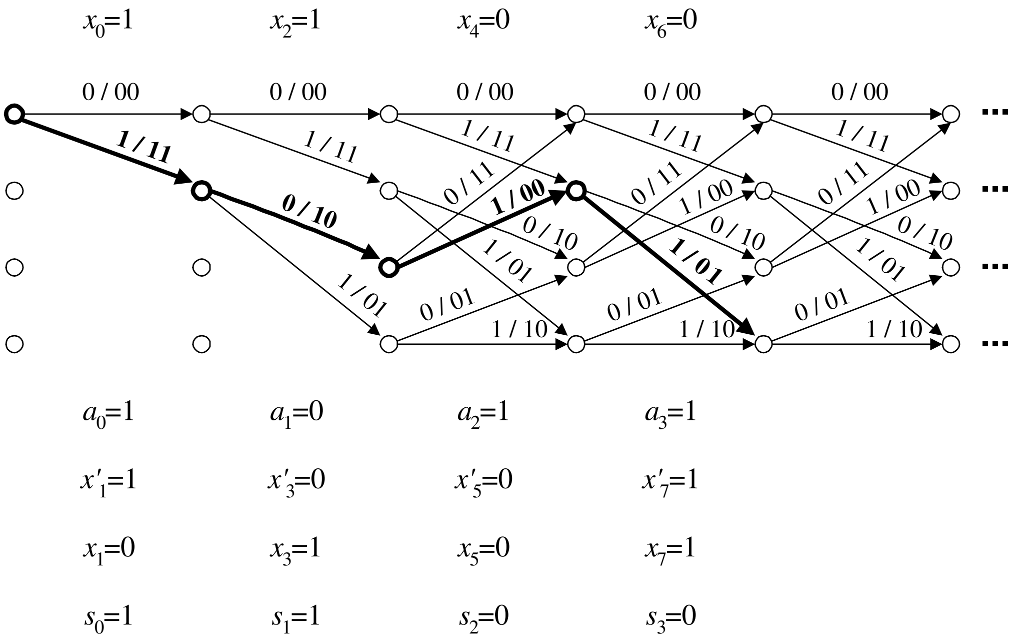

Fig. 11. The bits x_0, x_2, x_4, x_6 are used to determine the bold path through the encoding trellis of the linear convolutional code $[1 + D + D^2,\ 1 + D^2]$, i.e., a_0, a_1, a_2, a_6, exactly as in Fig. 10. The numbers over the transitions indicate the single input (information) bit and the two output (coded) bits when the trellis is used for channel encoding, i.e., they are in the form $a_i/x_{2i}x'_{2i+1}$. Based on these labels, the bits x'_1, x'_3, x'_5, x'_7 are determined by going through the selected path and then added to the rest of the bits of $\boldsymbol{x}$, i.e., x_1, x_3, x_5, x_7, to yield the corresponding syndrome bits s_0, s_1, s_2, s_3.

bits in $x'_k(D)$, $x'_{k+1}(D)$, $\ldots, x'_{n-1}(D)$ from the corresponding ones in $\boldsymbol{x}$, i.e., $s_i(D) = x'_{k+i}(D)+x_{k+i}(D)$, for $i = 0, 1, \ldots, (n-k-1)$. Fig. 11 shows how the path from Fig. 10 can yield the bits $x'_k(D)$, $x'_{k+1}(D), \ldots, x'_{n-1}(D)$ and how these bits can be combined with the respective bits in $x_k(D)$, $x_{k+1}(D), \ldots, x_{n-1}(D)$ to yield the syndrome bits $s_0(D)$, $s_1(D), \ldots, s_{n-k-1}(D)$.

The advantage of using the original channel encoding trellis defined by $\boldsymbol{G}(D)$ instead of the syndrome trellis defined by $\boldsymbol{H}(D)$ is that puncturing can easier be treated when looking at the $\boldsymbol{G}(D)$ trellis. Fig. 12 shows how when the second coded bit is punctured every second trellis step in the linear convolutional code $[1 + D + D^2,\ 1 + D^2]$, the path can be determined. From this path, the nonpunctured coded bits that were not used to determine the path are used to generate the corresponding syndrome bits. This approach can easily be generalized to any pattern of puncturing, random or periodic, as long as the nonpunctured bits can determine a unique path through the convolutional code trellis.

We will denote the above convolutional syndrome former with a block with input the length N bit sequence $\boldsymbol{x}$ and outputs the length $(N-K)$ syndrome bit sequence $\boldsymbol{s}$ and the corresponding length K information bit sequence $\boldsymbol{a}$, as shown in Fig. 13. The notation of Fig. 13 makes clear the difference with the nonsyndrome approaches of [1, 2, 3, 14]. The syndrome generation with the method presented here and any of the methods in [1, 2, 3, 14] can be considered equivalent. However, the main difference lies in the second output, the corresponding information sequence $\boldsymbol{a}$, which should be the only thing interleaved to preserve the turbo structure in the next section, unlike the approaches in [1, 2, 3, 14], where the input sequence $\boldsymbol{x}$ is interleaved.

Decoding: There are two equivalent ways to perform the decoding, assuming that the encoding was done as explained above. In both cases the side information sequence $\boldsymbol{y} = [y_0, y_1, \ldots, y_{N-1}]$ with length $N = n\,m$ is first used to form n different length m subsequences $\boldsymbol{y}_i = [y_i, y_{i+n}, \ldots, y_{i+n(m-1)}]$, $i = 0, 1, \ldots, n-1$. Each of these subsequences can be written in polynomial form as

$$y_i(D) = y_i + y_{i+n}\,D + y_{i+2n}\,D^2 + \ldots + y_{i+n(m-1)}\,D^{m-1} \tag{2.20}$$

The decoding for the codeword of $\boldsymbol{x}$ of the coset convolutional code indexed by the syndrome sequence $\boldsymbol{s}$, given that the side information $\boldsymbol{y}$ is correlated with $\boldsymbol{x}$, can be performed using the trellis of the linear binary convolutional code $\boldsymbol{G}(D)$. This can

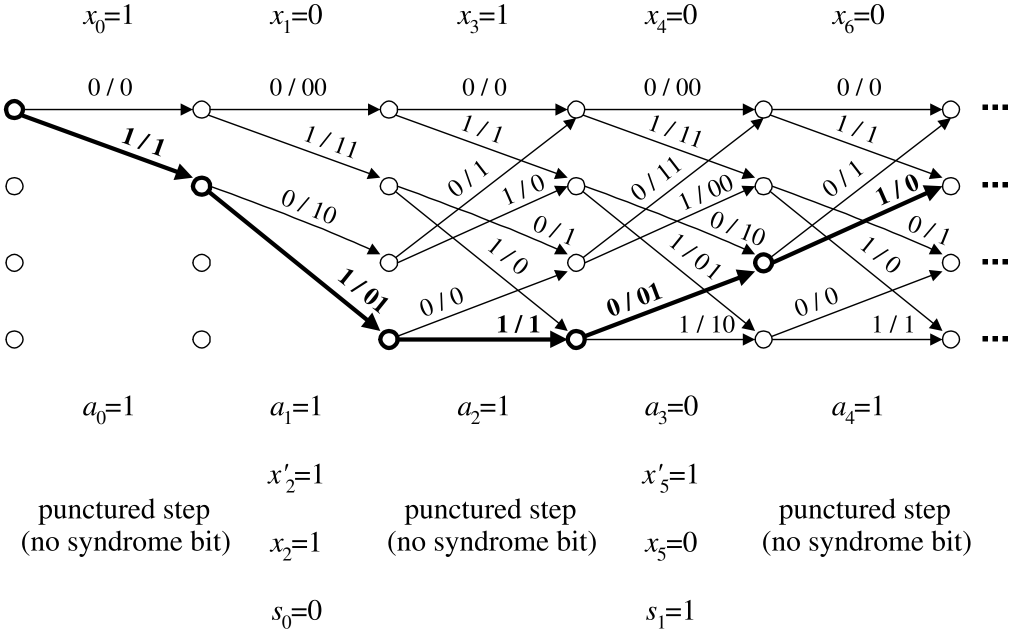

Fig. 12. The syndrome bits of the same bit sequence $x_0, x_1, x_2, x_3, \ldots, x_6$ as the one used in Fig. 11 are determined with respect to the punctured to rate $\frac{2}{3}$ linear convolutional code $[1 + D + D^2,\ 1 + D^2]$. The bits x_0, x_1, x_3, x_4, x_6 are used to determine the bold path through the encoding trellis of the punctured convolutional code, i.e., a_0, a_1, a_2, a_3, a_4. The numbers over the nonpunctured transitions indicate the single input (information) bit and the two output (coded) bits when the trellis is used for channel encoding, i.e., they are in the form $a_{2i+1}/x_{3i+1}x'_{3i+2}$. The numbers over the punctured transitions indicate the single input (information) bit and the single output (coded) bit when the trellis is used for channel encoding, i.e., they are in the form a_{2i}/x_{3i}. Based on these labels, the bits x'_2, x'_5 are determined by going through the selected path and then added to the rest of the bits of $\boldsymbol{x}$, i.e., x_2, x_5, to yield the corresponding syndrome bits s_0, s_1.

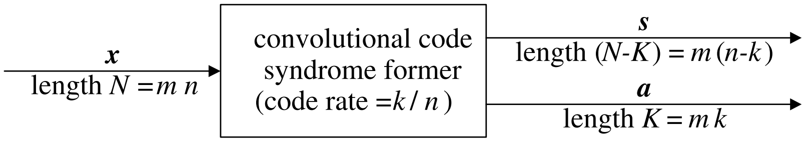

Fig. 13. The convolutional code syndrome former as a block with one input and two outputs: the input is the length N sequence $\boldsymbol{x}$ and the outputs are the length $(N-K)$ syndrome sequence $\boldsymbol{s}$ and the length K information sequence $\boldsymbol{a}$.

be done in one of the following two ways.

The first way is two modify the trellis of the linear convolutional code in the trellis steps where at least one of the $(n-k)$ $s_i(D)$'s are nonzero. We consider the j^{th} trellis step to correspond to the exponent j in the polynomials of D. There are $(n-k)$ polynomials $s_i(D)$ in D so if at least one of them includes a D^j term then the j^{th} trellis step has to account for that. If $s_i(D)$ is the only polynomial with a D^j term then the $(k+i)^{\text{th}}$ bit over all the transitions in the j^{th} trellis step has to be flipped. If more than one of the polynomials $s_i(D)$ have a D^j term, then more than one bits have to flipped at the corresponding positions of the j^{th} trellis step.

As an example consider the syndrome sequence formed in Fig. 11. Fig. 14 shows how this syndrome sequence can be used to modify the linear convolutional code trellis. Wherever a syndrome bit is not zero, the corresponding parity bit in all the labels of the respective trellis step has to be changed.

In the above way the coset convolutional code trellis is actually considered instead of the linear convolutional code trellis and any decoding algorithm for convolutional codes, e.g., the Viterbi or the BCJR algorithm, can be used on this modified trellis. In this approach the syndrome sequence was used to convert the trellis of the linear convolutional code into the trellis of the coset convolutional code by flipping the marked bits. The decoder outputs the most likely sequence $\hat{\boldsymbol{x}}$ of this modified trellis.

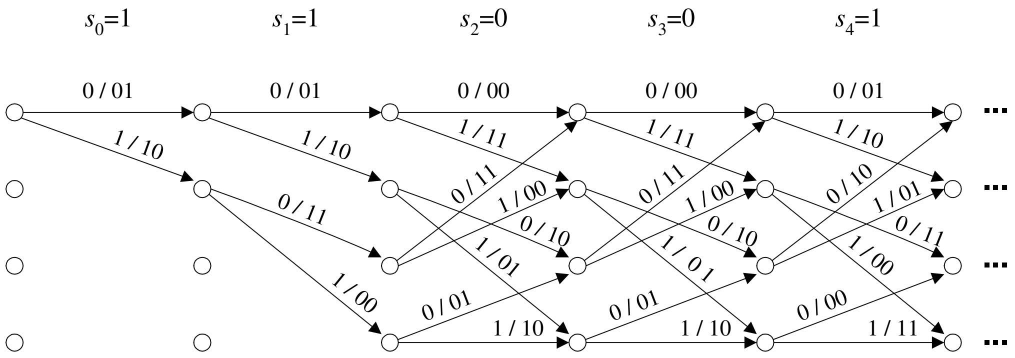

Fig. 14. The convolutional code syndromes are used to modify the convolutional code trellis used at the decoder.

An equivalent way of performing the above decoding without modifying the trellis structure of the linear convolutional code is to decode for the most likely linear convolutional codeword assuming that $\boldsymbol{y'} = [y'_0, y'_1, \ldots, y'_{N-1}]$ is correlated more with one of these codewords. $\boldsymbol{y'}$ is determined by combining the side information sequence $\boldsymbol{y}$ and the syndrome sequence $\boldsymbol{s}$ in the following manner.

$$y'_i(D) = \begin{cases} y_i(D) & \text{if } \ 0 \leq i < k \\ \\ y_i(D) + s_{i-k}(D) & \text{if } \ k \leq i < n \end{cases} \tag{2.21}$$

where

$$y'_i(D) = y'_i + y'_{i+n}\, D + y'_{i+2n}\, D^2 + \ldots + y'_{i+n(m-1)}\, D^{m-1} \tag{2.22}$$

This second approach amounts to first accounting for the coset offset in the side information $\boldsymbol{y}$ converting it into $\boldsymbol{y'}$ and then decoding the linear convolutional code for the most likely codeword $\hat{\boldsymbol{x}}'$. After the decoding the coset offset should be added back to form the most likely coset codeword $\hat{\boldsymbol{x}}$ as

$$\hat{x}_i(D) = \begin{cases} \hat{x}'_i(D) & \text{if } \ 0 \leq i < k \\ \\ \hat{x}'_i(D) + s_{i-k}(D) & \text{if } \ k \leq i < n \end{cases} \tag{2.23}$$

where

$$\hat{x}'_i(D) = \hat{x}'_i + \hat{x}'_{i+n}\, D + \hat{x}'_{i+2n}\, D^2 + \ldots + \hat{x}'_{i+n(m-1)}\, D^{m-1} \tag{2.24}$$

and similarly

$$\hat{x}_i(D) = \hat{x}_i + \hat{x}_{i+n}\, D + \hat{x}_{i+2n}\, D^2 + \ldots + \hat{x}_{i+n(m-1)}\, D^{m-1} \tag{2.25}$$

c. Concatenated Interleaved Codes

The main issue that should be accounted for in computing syndromes for concatenated interleaved codes is that the two component codes are connected through one of the two information sequences involved. E.g., in parallel concatenation the information sequence of one component code is the interleaved version of the information sequence of the other component code, while in serial concatenation only the information sequence of the inner component code is passed through the interleaver.

Therefore, the syndrome formers we considered in the previous section for convolutional codes come in handy, as they also include the generation of the corresponding information sequences. This is what makes this approach the only one so far to follow Wyner's scheme. In [1, 2, 3, 14] the whole sequence to be compressed is interleaved instead of the transformation of part of it. Therefore, the approach presented here corresponds to conventional concatenated interleaved codes employed in channel coding, while the schemes in [1, 2, 3, 14] do not seem to have an equivalent channel coding scheme.

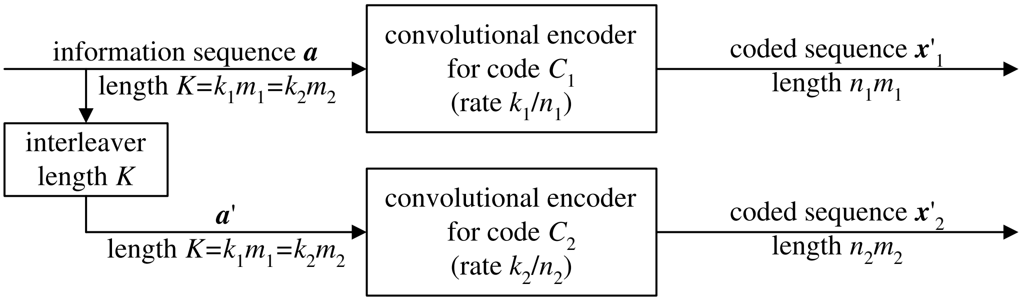

Fig. 15. The parallel concatenation of two interleaved convolutional codes in conventional channel coding.

We assume we have two possibly punctured convolutional component codes C_1 and C_2 of rates $\frac{k_1}{n_1}$ and $\frac{k_2}{n_2}$, respectively, and consider forming the syndromes of a binary sequence $\boldsymbol{x}$ of length N with respect to their parallel concatenation and their serial concatenation through interleavers. The code rate of the concatenated code is assumed to be $\frac{k}{n}$. We assume $\frac{k_1}{n_1} \leq 1$ and that there are k_1 independent columns out of the n_1 columns in the generator matrix of C_1, so that the syndrome forming approach of the previous section can be followed for the component code C_1, but the approach can be extended to the case of $\frac{k}{n} \leq 1$ in the parallel concatenation case. For the serial concatenation, we have to make the same assumption of the k_2 independent columns out of the n_2 columns in the generator matrix of C_2 as well. The overall approach can also be extended to the case of component block codes.

In the parallel (turbo) concatenation (Fig. 15), we assume that to generate a length $N = m\,n$ binary codeword, the information sequence $\boldsymbol{a}$ must have length $K = m\,k$, where m is a positive integer. This is also the interleaver length and it satisfies $K = m_1\,k_1 = m_2\,k_2$, where m_1 and m_2 are two positive integers. In this case, $m_1\,n_1$ coded bits are generated by the channel encoder of C_1 and $m_2\,n_2$ coded bits are generated by the channel encoder of C_2 and thus, the overall code rate can be

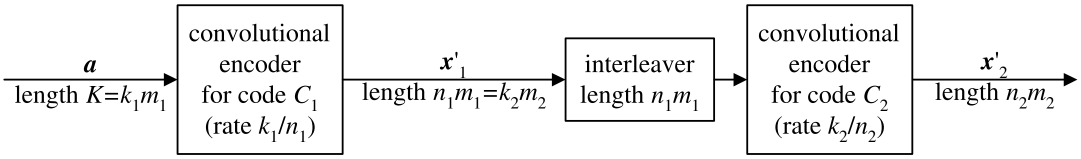
$\boldsymbol{a}$
length $K{=}k_1m_1$
convolutional encoder for code C_1 (rate k_1/n_1)
$\boldsymbol{x}'_1$
length $n_1m_1{=}k_2m_2$
interleaver length n_1m_1
convolutional encoder for code C_2 (rate k_2/n_2)
$\boldsymbol{x}'_2$
length n_2m_2

Fig. 16. The serial concatenation of two interleaved convolutional codes in conventional channel coding.

expressed as

$$\frac{k}{n} = \frac{K}{m_1\,n_1 + m_2\,n_2} \tag{2.26}$$

In the serial concatenation (Fig. 16), we assume that the code C_2 is the inner code [23]. So, to generate a length $N = m\,n = m_2\,n_2$ binary codeword, the information sequence $\boldsymbol{a}$ must have length $K = m\,k = m_1\,k_1$, where m, m_1 and m_2 are positive integers. The interleaver length in this case must be $m_1\,n_1 = m_2\,k_2$ and the overall rate can be expressed as

$$\frac{k}{n} = \frac{k_1\,k_2}{n_1\,n_2} \tag{2.27}$$

Showing how the syndrome formers of parallel and serial concatenated interleaved codes with two component codes operate, the approach in this subsection can be extended to any hybrid concatenation, i.e., any combination of parallel and serial concatenation, of more than two codes.

Encoding: In the parallel concatenation, we form the $(N - K)$ long syndrome sequence of a binary sequence $\boldsymbol{x}$ of length N. Since $N = m_1\,n_1 + m_2\,n_2$ with $m_1\,n_1 \geq K = m_1\,k_1$, the first $m_1\,n_1$ bits of the sequence $\boldsymbol{x}$ of length N are used to determine $(m_1\,n_1 - K)$ syndrome bits with respect to the code C_1, as explained in the previous part on convolutional codes. Following the approach of the previous part on convolutional codes, the code C_1 is also used to determine the information

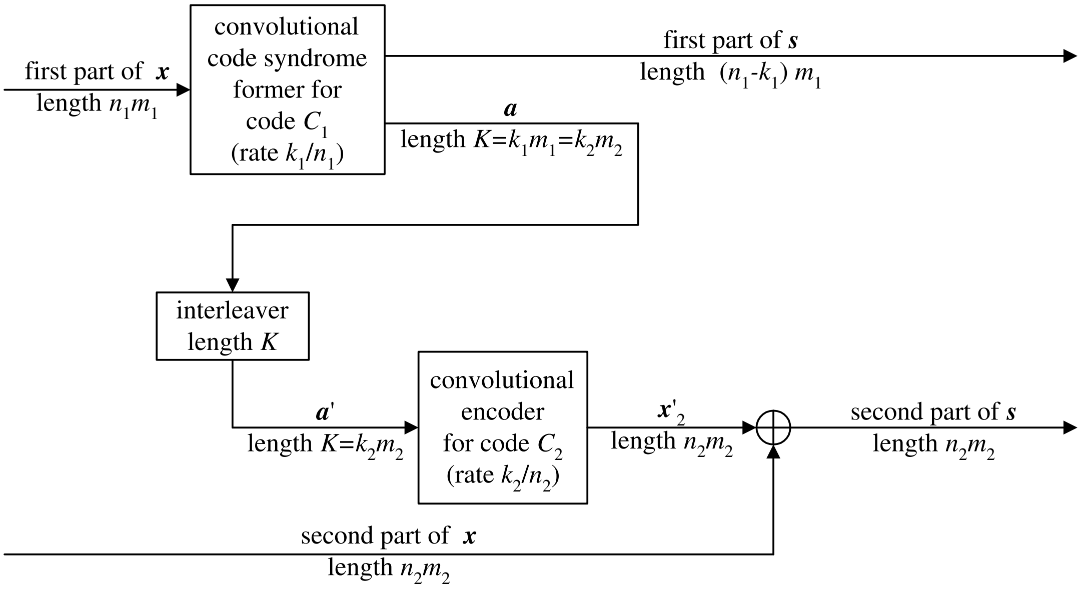

Fig. 17. The syndrome former for the parallel concatenation of two interleaved convolutional codes.

sequence $\boldsymbol{a}$.

The second step of the parallel concatenated syndrome former is to interleave the information sequence $\boldsymbol{a}$. The interleaved information sequence $\boldsymbol{a}'$ can be passed to the second component code to determine the corresponding $m_2\,n_2$ coded bits. These bits are added to the remaining $m_2\,n_2$ bits of the sequence $\boldsymbol{x}$ to form the remaining $m_2\,n_2$ syndrome bits. This approach is shown in Fig. 17.

In the case of serial concatenation, the first $m_2(n_2 - k_2)$ syndrome bits of the total $(N - K) = (m_2\,n_2 - m_1\,k_1)$ syndrome bits are determined with respect to the inner code C_2, as explained in the previous previous part on convolutional codes. The corresponding length $m_1\,k_1$ information sequence $\boldsymbol{a}$ of the inner code C_2 is also determined.

Then this information sequence $\boldsymbol{a}$ is interleaved to form $\boldsymbol{a}'$. The remaining

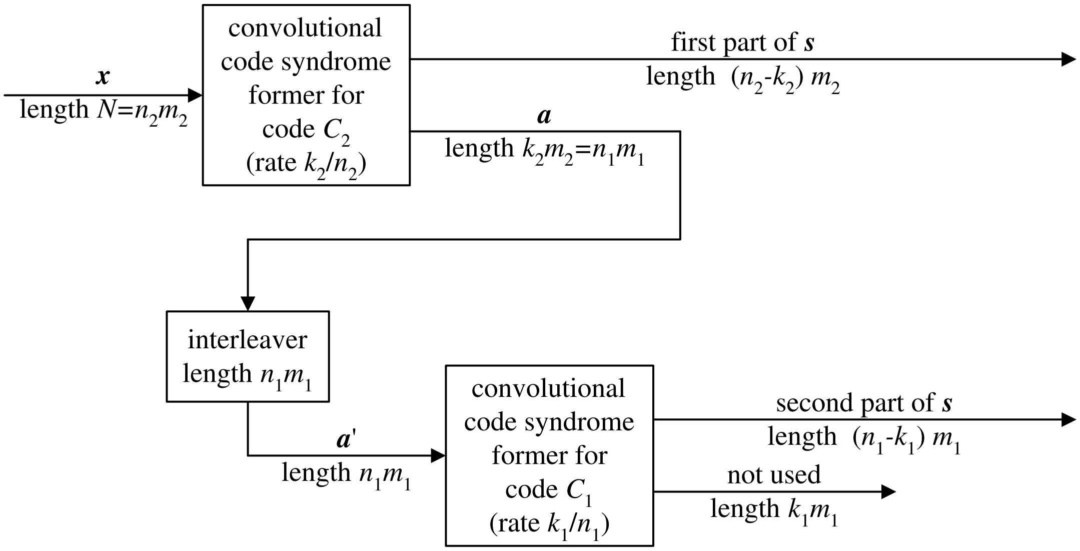

Fig. 18. The syndrome former for the serial concatenation of two interleaved convolutional codes.

$m_1(n_1-k_1) = (m_2\,k_2 - m_1\,k_1)$ syndrome bits are determined as the syndromes of the sequence $\boldsymbol{a}'$ with respect to the outer code C_1. This approach is shown in Fig. 18.

Decoding: Decoding is similar to decoding of convolutional codes over each component coset code. The syndrome bits are split into two subsets corresponding to the two component codes. Both approaches of either modifying the respective trellis to form the coset code trellis, or adding the syndrome bits before and after decoding and using the linear code trellis can be applied.

We should just note that in the case of the parallel concatenation the second component code has all its coded bits modified by the respective syndrome bits and if the first component code is systematic and thus the whole turbo code is systematic, the second component decoder also uses the side information bits corresponding to the systematic bits in decoding.

Scaling and/or clipping of the extrinsic information exchanged between the component decoders in the iterative algorithm is also helpful, as first pointed out in [24].

d. Simulation Results[†]

We first simulated left regular ($\lambda(x)$ has a single term) LDPC codes and compared the results with those of nonsyndrome turbo-codes [1, 2] using approximately the same parameters. The codeword length is $n = 16384$, i.e., equal to the interleaver length of [1, 2].[2] More than 2000 blocks were transmitted without a single error after 40 iterations of the message passing algorithm. As for the LDPC code distribution, $\lambda(x) = x^2$ and $\rho(x) = (1-\rho_r)x^{r-2}+\rho_r x^{r-1}$, where $\bar{r} = (1-\rho_r)(r-1)+\rho_r r = r-1+\rho_r$ varies so that the code rate changes. The results are given in Table II together with those of [1, 2]. R_1 and R_2 are the rates in compressed bits per information bit used for X_i and Y_i, respectively and so in our case always $R_2 = 1$. From the table, it is clear that even these simple LDPC codes can exhibit small gains over the turbo code performance.

In Table II, we also compare the conventional turbo code design with the non-conventional of [1, 2] using the same uncompressed input sequence length $n = 16384$ and the same generator polynomial $G(D) = (1+D+D^2+D^3)/(1+D^2+D^3)$, i.e., the first component code C_1 is the systematic convolutional code $[1, 17/13]$ and the second component code C_2 is the convolutional code $[17/13]$. Because of the fixed $n = 16384$,

[†] ©2002,2003 IEEE. Reprinted, with permission, from "Compression of binary sources with side information at the decoder using LDPC codes", Angelos D. Liveris, Zixiang Xiong, and Costas N. Georghiades, *IEEE Commun. Letters*, Oct. 2002, pp. 440-442 and from "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes", Angelos D. Liveris, Zixiang Xiong, and Costas N. Georghiades, in *Proc. 2003 IEEE Data Compression Conference (DCC '03)*, Mar. 2003, pp. 193-202.

[2] If we wish to impose the additional requirement of symmetric rates ($R_1 = R_2$), we should use time-sharing with $n = 8192$ for a fair comparison with [1], which will slightly worsen the LDPC code performance.

Table II. Comparison of [1, 2] with our LDPC and Conventional Turbo Code Results

p	0.05	0.1
$H(X,Y)$ $(1+H(p))$	1.286	1.469
R_1+R_2 [1]	1.435	1.630
R_1+1 [2]	1.380	1.580
R_1+1 (LDPC)	1.402	1.600
R_1+1 (conventional turbo)	1.388	1.556

the puncturing and the interleaver length in our conventional codes has to be different each time to guarantee $n \leq 16384$. In conventional systematic turbo codes, only the parity bits are punctured. Following this approach here, and as less parity bits means less syndrome bits, and thus higher compression, the two component convolutional source encoders use the punctured trellis of the original convolutional channel code.

For instance, for $p = 0.1$, the orginal turbo channel code would yield 5 parity bits for every 4 input bits, i.e., 3 parity bits from the first component code and 2 from the second. Hence, there would be $4+5=9$ turbo coded bits resulting in an overall channel code rate of $4/9 = 0.444$. The corresponding turbo source encoder would take 9 input bits and output 5 syndrome bits, 3 from the first component code and 2 from the second, achieving 9:5 compression ($R_1 = 0.556$).

For $p = 0.05$, 0.1, the interleaver length was 10020 and 7280, respectively. Both interleavers were s-random with spread $s = 42$, 36, respectively. Our results in Table II correspond to zero errors after 15 decoding iterations for 2500 blocks.

Much better results are expected from LDPC codes with improved irregular code design [18, 19, 20, 21]. We use the simplest good irregular code of [20],[3] i.e., the rate

[3] The irregular codes in [20] have been optimized for the additive white Gaussian noise (AWGN) channel, but they are expected to perform better than regular or nonoptimized irregular codes over the binary symmetric channel (BSC) as well [20].

1/2 code with the degree distribution:

$$\begin{aligned}\lambda(x) &= 0.234029x + 0.212425x^2 + 0.146898x^5 \\ &\quad + 0.102840x^6 + 0.303808x^{19} \qquad (2.28) \\ \rho(x) &= 0.71875x^7 + 0.28125x^8 \qquad (2.29)\end{aligned}$$

This irregular code was simulated together with the regular (3,6) code in Fig. 19. The codeword (frame) length was $n = 10^4$ for the regular code and $n = 10^4$ and $n = 10^5$ for the irregular code (marked "awgn"). The bit error rate (BER) for X_i, $\Pr[\hat{X}_i \neq X_i]$, was measured after 100 iterations of the decoding algorithm. The Slepian-Wolf theoretical limit of 0.5 bits and the best nonsyndrome turbo code performance reported for this code rate [3] (about 0.38 bits for codeword length 10^5) are also shown in Fig. 19. The second irregular code, marked "bsc" in Fig. 19, is the rate 1/2 LDPC code given in example 2 of [19], which has been optimized for the binary symmetric channel (BSC). The threshold for this code is $H(p^*) = 0.488$ bits [19], also shown in Fig. 19.

In Fig. 19 even the regular (3,6) code slightly outperforms the turbo coding schemes of [3, 14], which use greater or equal codeword length, while the irregular code of (2.28) and (2.29) almost halves the gap to the Slepian-Wolf limit, even with a codeword length of $n = 10^4$. Further irregular LDPC code design for the binary symmetric channel yields slightly better results as the difference between the "awgn" and "bsc" curves of Fig. 19 shows.

For larger interleaver lengths, the advantage of conventional over nonconventional turbo schemes is also expected to be clearer. This can be seen in Fig. 19, where serial and parallel concatenated codes of rate 1/2 have also been simulated. The serial concatenation has the $[1, 5/7]$ convolutional code as the outer code and the differential encoder [1/3] as the inner code. The parallel concatenated code of Fig. 19 is the best punctured turbo code of memory 4 in [25] with generator polynomial $G(D) = (1 + D + D^4)/(1 + D^3 + D^4)$, which gives very good performance even for severe puncturing. The block size in each of the serial and parallel concatenation cases refers to the overall codeword size, which is equal to the interleaver size only for the serial concatenation. For the parallel case the interleaver length was 50000. All interleavers were s-random, 30 iterations were used in the decoder and for each simulated point either 100 block errors or $5 \cdot 10^8$ bits were simulated.

In Fig. 19, the Slepian-Wolf limit is given as a special case of the Wyner-Ziv limit [26], i.e., for practically lossless compression (BER for X_i =Pr$[\hat{X}_i \neq X_i] \leq 10^{-6}$). As in conventional channel coding, in Fig. 19, conventional parallel and serial concatenated interleaved codes outperform regular low-density parity-check (LDPC) codes but not optimized irregular LDPC codes. Furthermore, parallel concatenation can converge much faster. Whichever way we employ, however, to concatenate the binary convolutional codes, as long as it is done conventionally, the performance is better than nonconventional turbo schemes as shown in Fig. 19, where the performance of [3] is also included (codeword length in the order of 10^5), and very close to the theoretical limits.

However, regular LDPC codes could not outperform the codes of [3] for higher compression, i.e., for 4:1 and 8:1 compression only good irregular codes could achieve slightly higher compression. In Table III our results for length 10^5 irregular LDPC codes are compared with those of [3] (length 10^5). For the rate 1/2 codes we considered the results of the LDPC code of example 2 of [19], whereas for the other two cases we designed a rate 3/4 and a rate 7/8 LDPC code respectively using the method of

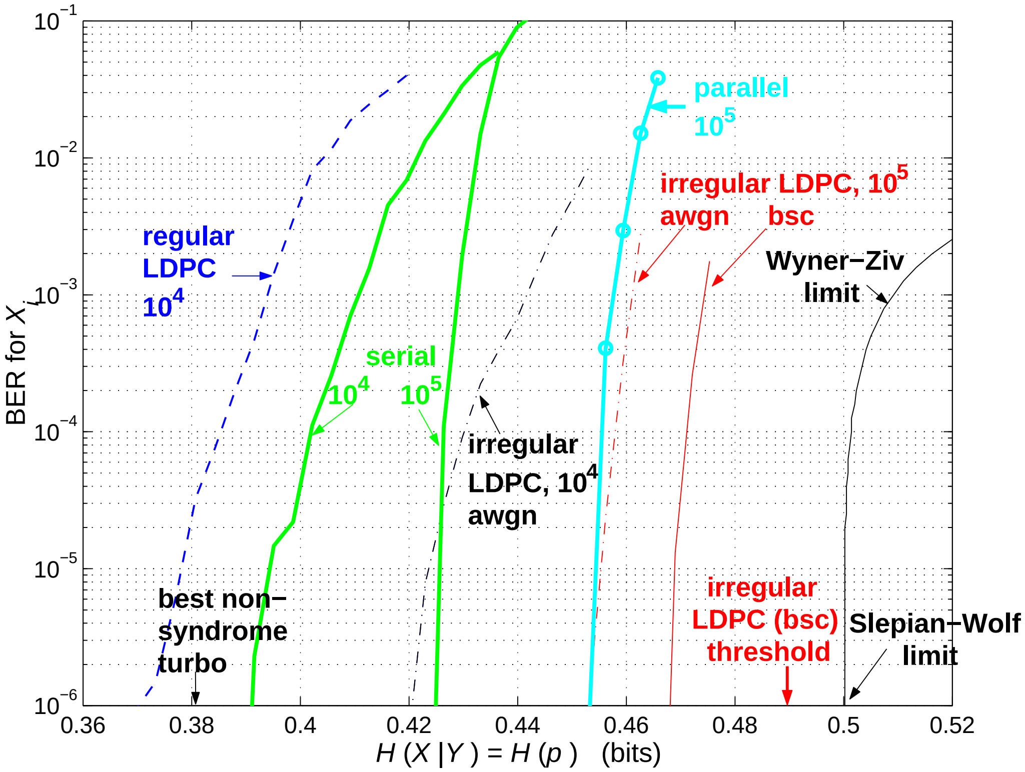

Fig. 19. Conventional serial and parallel concatenated interleaved convolutional codes compared with regular and irregular LDPC and nonconventional turbo codes [3]. The theoretical limits are also shown.

[18], obtaining the distributions

$$\lambda(x) = 0.071428x + 0.230118x^2 + 0.079596x^9 + 0.147043x^{10} + 0.073821x^{48} + 0.397994x^{49}, \tag{2.30}$$

$$\rho(x) = x^{27} \tag{2.31}$$

for $R_1 = 0.25$ and

$$\lambda(x) = 0.034482x + 0.270427x^2 + 0.027719x^9 + 0.209427x^{10} + 0.457945x^{49}, \tag{2.32}$$

$$\rho(x) = x^{57} \tag{2.33}$$

for $R_1 = 0.125$. For the design of these two LDPC codes we considered right regular codes and allowed a maximum left node degree of 50. To easily avoid length 4 cycles of degree-2 variable nodes, we also upper-bounded the percentage of degree-2 variable nodes. All irregular LDPC results shown in Table III assume a probability of error smaller than 10^{-6} for $5 \cdot 10^8$ simulated bits and 100 iterations.

In Table III, the results for conventional turbo codes for higher compression ratios are also given for $\Pr[\hat{X}_i \neq X_i] \leq 10^{-6}$ with 30 iterations at codeword length 10^5 (interleaver size 87500, 75000 and 50000 respectively, always s-random interleaver) and $5 \cdot 10^8$ simulated bits. From both Tables II and III, it turns out that, as expected, the schemes of [2, 3] exhibit similar performance to the conventional codes for severe puncturing, but the conventional approach is clearly better for less puncturing.

2. Symmetric Coding

The asymmetric binary Slepian-Wolf problem, also referred to as lossless source coding with side information at the decoder, was the focus of the previous subsection.

Table III. Comparison of [3] with our LDPC and Conventional Turbo Code Results

R_1	0.125	0.250	0.500
$H(X\|Y)$ [3]	0.089	0.189	0.381
$H(X\|Y)$ (LDPC)	0.091	0.204	0.466
$H(X\|Y)$ (conventional turbo)	0.084	0.194	0.453

Its correspondence to channel coding has been established [16, 17, 27] and Slepian-Wolf limit approaching codes based on this correspondence were designed for the binary setup in the previous subsection. This approach will also be further extended to the asymmetric Slepian-Wolf problem of more than two sources [28] in the next subsection.

The way asymmetric Slepian-Wolf coding can be used for Slepian-Wolf coding of two or more correlated sources is through time-sharing. However, sometimes time-sharing might not be desirable or practical and *symmetric* Slepian-Wolf coding might be needed. Symmetric Slepian-Wolf coding refers to the case of having both sources coded instead of one and the other one available as side information as in the case asymmetric Slepian-Wolf coding. Some first approaches of Slepian-Wolf coding have been first suggested in [1, 29], with [29] trying to provide a more general setup for binary systematic codes.

The purpose of this subsection is to clarify this general binary setup of [29] so that advanced channel codes can be employed to yield good symmetric binary Slepian-Wolf codes. We first start with the systematic setup of [29] and continue with practical code designs based on systematic turbo and irregular repeat-accumulate (IRA) codes [30].

a. Systematic Codes

To simplify things we start with the completely symmetric Slepian-Wolf scenario, i.e., both encoders use the same rate. However, the generalization to any rate allocation between the sources and to any number of sources is straightforward. In fact the asymmetric Slepian-Wolf scenario of the previous subsection is a special case of this symmetric scenario.

As in [29], we start with an (n, k) binary channel code $\mathcal{C}$, whose $k \times n$ generator matrix $\boldsymbol{G}$ is

$$\boldsymbol{G} = [\boldsymbol{I}_k \quad \boldsymbol{P}] \tag{2.34}$$

$\boldsymbol{I}_k$ is a $k \times k$ identity matrix and $\boldsymbol{P}$ is a $k \times (n-k)$ matrix. For completely symmetric Slepian-Wolf, we assume that k is an even number.

Let $m = \frac{k}{2}$. Two (n, m) subcodes $\mathcal{C}_1$ and $\mathcal{C}_2$ of $\mathcal{C}$ can be defined with generator matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$, respectively,

$$\boldsymbol{G}_1 = [\boldsymbol{I}_m \quad \boldsymbol{O}_{m,m} \quad \boldsymbol{P}_1] \tag{2.35}$$

$$\boldsymbol{G}_2 = [\boldsymbol{O}_{m,m} \quad \boldsymbol{I}_m \quad \boldsymbol{P}_2] \tag{2.36}$$

where $\boldsymbol{O}_{m,m}$ is an $m \times m$ all-zeros matrix and $\boldsymbol{P}^T = \begin{bmatrix}\boldsymbol{P}_1^T & \boldsymbol{P}_2^T\end{bmatrix}$. It is easy to see that if $\boldsymbol{c}_1$ is any codeword of $\mathcal{C}_1$ and $\boldsymbol{c}_2$ is any codeword of $\mathcal{C}_2$ then $\boldsymbol{c}_1 + \boldsymbol{c}_2$ is always a codeword of $\mathcal{C}$, where "+" means addition in GF(2).

Two simple choices for the $(n - m) \times n$ parity-check matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ of $\mathcal{C}_1$

and $\mathcal{C}_2$, respectively, can be shown to be

$$\boldsymbol{H}_1 = \begin{bmatrix} \boldsymbol{O}_{m,m} & \boldsymbol{I}_m & \boldsymbol{O}_{m,n-k} \\ \boldsymbol{P}_1^T & \boldsymbol{O}_{n-k,m} & \boldsymbol{I}_{n-k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{O}_{m,m} & \\ \boldsymbol{P}_1^T & \end{bmatrix} \boldsymbol{I}_{n-m} \Bigg] \tag{2.37}$$

$$\boldsymbol{H}_2 = \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O}_{m,m} & \boldsymbol{O}_{m,n-k} \\ \boldsymbol{O}_{n-k,m} & \boldsymbol{P}_2^T & \boldsymbol{I}_{n-k} \end{bmatrix} \tag{2.38}$$

since both $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ have rank $n-m$ and $\boldsymbol{H}_1\boldsymbol{G}_1^T = \boldsymbol{H}_2\boldsymbol{G}_2^T = \boldsymbol{O}_{n-m,m}$.

To use these subcodes for totally symmetric SW, we have to form syndromes for each of the binary n-tuples $\boldsymbol{x}$ and $\boldsymbol{y}$ at the output of each source. To do so, we write $\boldsymbol{x}$ and $\boldsymbol{y}$ in the form

$$\boldsymbol{x} = [\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \boldsymbol{q}_1] \tag{2.39}$$

$$\boldsymbol{y} = [\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \boldsymbol{q}_2] \tag{2.40}$$

where $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ are all row-vectors of length m and $\boldsymbol{q}_1$, $\boldsymbol{q}_2$ are row-vectors of length $n-k=n-2m$.

The length $n-m$ syndromes $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ formed by the two subcodes are

$$\boldsymbol{s}_1^T = \boldsymbol{H}_1\boldsymbol{x}^T = \begin{bmatrix} \boldsymbol{u}_2^T \\ \boldsymbol{P}_1^T\boldsymbol{u}_1^T + \boldsymbol{q}_1^T \end{bmatrix} \tag{2.41}$$

$$\boldsymbol{s}_2^T = \boldsymbol{H}_2\boldsymbol{y}^T = \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{P}_2^T\boldsymbol{v}_2^T + \boldsymbol{q}_2^T \end{bmatrix} \tag{2.42}$$

We define the length n row-vectors $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ as

$$\boldsymbol{t}_1^T = \begin{bmatrix} \boldsymbol{O}_{m,1} \\ \boldsymbol{u}_2^T \\ \boldsymbol{P}_1^T \boldsymbol{u}_1^T + \boldsymbol{q}_1^T \end{bmatrix} \tag{2.43}$$

$$\boldsymbol{t}_2^T = \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{O}_{m,1} \\ \boldsymbol{P}_2^T \boldsymbol{v}_2^T + \boldsymbol{q}_2^T \end{bmatrix} \tag{2.44}$$

Then the length n row-vectors $\boldsymbol{x} + \boldsymbol{t}_1$ and $\boldsymbol{y} + \boldsymbol{t}_2$ are codewords of the codes $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively.

So by sending $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ from the two encoders to the joint decoder, the decoder finds the codeword in $\mathcal{C}$ that is closest to $\boldsymbol{t}_1 + \boldsymbol{t}_2$, since the binary words $\boldsymbol{x}$ and $\boldsymbol{y}$ are correlated through the binary symmetric model and thus, with high probability their sum $\boldsymbol{x} + \boldsymbol{y}$ has small weight. If there is no error in decoding, this codeword should be $\boldsymbol{x} + \boldsymbol{t}_1 + \boldsymbol{y} + \boldsymbol{t}_2$. The corresponding $\hat{\boldsymbol{u}}_1$ and $\hat{\boldsymbol{v}}_2$ are then known as the systematic part of the codeword. But $\boldsymbol{u}_1 \boldsymbol{G}_1 = \boldsymbol{x} + \boldsymbol{t}_1$ and so

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{u}}_1 \boldsymbol{G}_1 + \boldsymbol{t}_1 \tag{2.45}$$

Similarly

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{v}}_2 \boldsymbol{G}_2 + \boldsymbol{t}_2 \tag{2.46}$$

So if the binary code $\mathcal{C}$ is approaching the capacity of the BSC, when used as above this code will approach the Slepian-Wolf limit in the binary symmetric setup, as long as the correlation between $\boldsymbol{x}$ and $\boldsymbol{y}$ can be modelled with the same BSC. In other words, the above described approach is the extension of the syndrome scheme (Wyner's scheme) to the symmetric Slepian-Wolf scenario for systematic codes. Especially for the BSC correlation model, we should not expect performance differences when using either asymmetric or completely symmetric coding based on the same channel code.

A note about the difference of this approach to [1, 29]. [1] sends as above half of the systematic bits from the one encoder and half from the other encoder. But these systematic bits are used in encoding in [1], unlike the syndrome scheme above. But the main differences in [1] are first that parity bits instead of syndrome bits are sent and second that the two subcodes are quite separate (each subcode is a different turbo code forming a big turbo code with four component codes).

As for [29] which has inspired this work, the systematic approach is not so well handled in the sense that the component codes appear to be systematic but not the overall code $\mathcal{C}$. So this work is actually just a more clear and detailed formulation of what was suggested in [29], which is enough to allow us to extend the approach to symmetric Slepian-Wolf codes based on advanced channel codes.

It is clear from the above how by using $m_1 \neq m_2$ with $m_1 + m_2 = k$ instead of $m_1 = m_2 = m = \frac{k}{2}$ could allow any asymmetric rate. Furthermore, by using three subcodes with $m_1+m_2+m_3 = k$ or more, the same approach could be generalized to any number of sources and any asymmetric rate combination. This observation, made for the first time here, is perhaps the real advantage of this generalized symmetric approach, as a single code can be used to approach the joint entropy limit. However, there are some problems with the symmetric approach in the case of more than two sources, which will be address in more detail in the subsection on Slepian-Wolf coding of more than two sources.

b. Nonsystematic Codes

The above approach can also be extended to nonsystematic codes by expressing the nonsystematic (n, k) binary channel code $\mathcal{C}$ with its $k \times n$ generator matrix $\boldsymbol{G}$ in the form

$$\boldsymbol{G} = [\boldsymbol{A} \quad \boldsymbol{P}] \tag{2.47}$$

where $\boldsymbol{A}$ is a $k \times k$ invertible matrix and $\boldsymbol{P}$ is a $k \times (n-k)$ matrix. If the first k columns of the generator matrix $\boldsymbol{G}$ are not linearly independent, the columns of $\boldsymbol{G}$ can be rearranged so that the first k columns become linearly independent.

Then through $\boldsymbol{A}$ and its inverse, the same approach as before can be used, i.e., splitting the source output sequences $\boldsymbol{x}$ and $\boldsymbol{y}$ as in equations (2.39) and (2.40), respectively, the compressed sequences sent to the decoder have the form

$$\boldsymbol{s}_1^T = \begin{bmatrix} \boldsymbol{u}_2^T \\ \boldsymbol{P}^T \boldsymbol{A}^{-T} [\boldsymbol{u}_1 \quad \boldsymbol{O}_{1,m}]^T + \boldsymbol{q}_1^T \end{bmatrix} \tag{2.48}$$

$$\boldsymbol{s}_2^T = \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{P}^T \boldsymbol{A}^{-T} [\boldsymbol{O}_{1,m} \quad \boldsymbol{v}_2]^T + \boldsymbol{q}_2^T \end{bmatrix} \tag{2.49}$$

instead of (2.41) and (2.42), respectively, where the superscript $^{-T}$ denotes inverse transpose matrix.

At the decoder the vectors $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ are formed by inserting m zeros just after the m first bits of $\boldsymbol{s}_1$ and at the beginning of $\boldsymbol{s}_2$, respectively. Then decoding is done over the code C for the most likely codeword given $\boldsymbol{t}_1 + \boldsymbol{t}_2$. If no decoding error occurs, the decoded codeword of C is $\boldsymbol{x} + \boldsymbol{y} + \boldsymbol{t}_1 + \boldsymbol{t}_2$, from which knowing $\boldsymbol{v}_1$ and $\boldsymbol{u}_2$, $\boldsymbol{u}_1$ and $\boldsymbol{v}_2$ are determined and from these and $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$.

Another approach using a parity check matrix of the channel code C instead of the generator matrix of C used here was independently introduced in [31] for nonsystematic codes. However, this approach did not address some of the practical issues arising due to the matrix inversion, which are discussed next.

c. Practical Systematic Codes

It is straightforward to use any systematic practical codes, e.g., IRA codes, for completely symmetric Slepian-Wolf coding. For an (n,k) such code, the information part is split into two length $m = \frac{k}{2}$ parts. At the first encoder, the length n source output $\boldsymbol{x}$ is split into three parts in the form

$$\boldsymbol{x} \;=\; [\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \boldsymbol{q}_1] \tag{2.50}$$

where $\boldsymbol{u}_1$, $\boldsymbol{u}_2$ are row-vectors of length m and $\boldsymbol{q}_1$ is a row-vector of length $n-k = n-2m$.

To determine $\boldsymbol{s}_1$, the graph of the (n,k) IRA code is used and first $\boldsymbol{u}_1\boldsymbol{P}_1$ is determined by setting the values of the systematic IRA variable nodes equal to $[\boldsymbol{u}_1 \quad \boldsymbol{O}_{1,m}]$. In other words half of the systematic part is set equal to zero to determine $\boldsymbol{u}_1\boldsymbol{P}_1$.

But the length $n-m$ syndrome $\boldsymbol{s}_1$ that has to be formed by this first encoder is

$$\boldsymbol{s}_1^T \;=\; \boldsymbol{H}_1\,\boldsymbol{x}^T = \begin{bmatrix} \boldsymbol{u}_2^T \\ \boldsymbol{P}_1^T\boldsymbol{u}_1^T + \boldsymbol{q}_1^T \end{bmatrix} \tag{2.51}$$

So $\boldsymbol{q}_1$ has to be added to $\boldsymbol{u}_1\boldsymbol{P}_1$ and by appending $\boldsymbol{u}_2$ to $\boldsymbol{u}_1\boldsymbol{P}_1 + \boldsymbol{q}_1$, we get $\boldsymbol{s}_1$.

Similarly, $\boldsymbol{s}_2$ is formed at the second encoder. At the joint decoder, the same procedure is followed as described in the previous part on systematic codes. Again everything can be generalized to any rate allocation scenario with any number of sources. Furthermore, similarly, i.e., by setting half of the systematic bits equal to zero at each encoder, the same symmetric Slepian-Wolf coding approach can be extended to any systematic code, e.g., systematic turbo codes. Simulation results presented in [32] for systematic IRA and turbo codes show that indeed the code design works as well as expected.

For nonsystematic codes to avoid the complexity of inverting a matrix, convolutional and concatenated convolutional codes could be used, since, as in asymmetric Slepian-Wolf coding, the matrix inversion can be performed through the path selection on the code trellis. Parallel concatenation seems to be more suitable than serial concatenation in this symmetric coding setup, because the information sequence is passed through a single component code in parallel concatenated codes instead of two in the serial concatenation and thus, the inversion can be performed just through this single component code.

As a result, if a linear code is used that can approach the capacity of a BSC, then if the same BSC models the correlation between $\boldsymbol{x}$ and $\boldsymbol{y}$, the resulting Slepian-Wolf coding scheme based on the above setup will also approach the Slepian-Wolf limit for any rate allocation between the two encoders. When more than two sources are used, modelling the exact correlation model with a channel is more involved. Therefore, it might be more difficult to find good codes for symmetric SW coding of more than two sources, as discussed next.

3. Coding for More Than Two Sources

In this subsection, we will discuss how the Slepian-Wolf coding problem can be extended to more than two sources. We will first consider the binary symmetric model for three sources and then see how it can be generalized. Just for this subsection we will use the notation X_i, $i = 1, 2, \ldots, M$, for the output random variable of the i^{th} binary source, where M is the number of sources.

a. Three Sources

Assuming three binary sources with equiprobable outputs X_1, X_2, X_3 and $\Pr[X_2 \neq i|X_1 = i] = p$, $i = 0, 1$, imposing the symmetry condition, i.e., requiring that $\Pr[X_3 = i|X_1 = j, X_2 = k]$ $=$$\Pr[X_3 = 1-i|X_1 = 1-j, X_2 = 1-k]$, for all $i, j, k \in \{0, 1\}$, produces

$$\begin{aligned}
\Pr[X_3 = 0|X_1 = 0, X_2 = 0] &= a && (2.52)\\
\Pr[X_3 = 1|X_1 = 0, X_2 = 0] &= 1-a && (2.53)\\
\Pr[X_3 = 0|X_1 = 0, X_2 = 1] &= b && (2.54)\\
\Pr[X_3 = 1|X_1 = 0, X_2 = 1] &= 1-b && (2.55)\\
\Pr[X_3 = 0|X_1 = 1, X_2 = 0] &= 1-b && (2.56)\\
\Pr[X_3 = 1|X_1 = 1, X_2 = 0] &= b && (2.57)\\
\Pr[X_3 = 0|X_1 = 1, X_2 = 1] &= 1-a && (2.58)\\
\Pr[X_3 = 1|X_1 = 1, X_2 = 1] &= a && (2.59)
\end{aligned}$$

(2.52)-(2.59) result from just imposing the symmetry condition on all these binary conditional probabilities. However, this condition is so strong that they also satisfy both the uniform marginal probabilities for all three sources X_1, X_2, X_3 and the conditional probabilities $\Pr[X_2 \neq i|X_1 = i] = p$, $i = 0, 1$. So the model has actually three degrees of freedom (parameters) p, a, b.

This becomes clear when expressing the corresponding conditional probabilities

$\Pr[X_1 = i, X_2 = j | X_3 = k]$, $i = 0, 1$, $j = 0, 1$, $k = 0, 1$

$$\Pr[X_1 = 0, X_2 = 0 | X_3 = 0] = a(1-p) \tag{2.60}$$

$$\Pr[X_1 = 0, X_2 = 1 | X_3 = 0] = bp \tag{2.61}$$

$$\Pr[X_1 = 1, X_2 = 0 | X_3 = 0] = (1-b)p \tag{2.62}$$

$$\Pr[X_1 = 1, X_2 = 1 | X_3 = 0] = (1-a)(1-p) \tag{2.63}$$

$$\Pr[X_1 = 0, X_2 = 0 | X_3 = 1] = (1-a)(1-p) \tag{2.64}$$

$$\Pr[X_1 = 0, X_2 = 1 | X_3 = 1] = (1-b)p \tag{2.65}$$

$$\Pr[X_1 = 1, X_2 = 0 | X_3 = 1] = bp \tag{2.66}$$

$$\Pr[X_1 = 1, X_2 = 1 | X_3 = 1] = a(1-p) \tag{2.67}$$

There is a way to reduce the degrees of freedom from three to one and simplify the whole problem by imposing even stronger symmetry, i.e., require both $\Pr[X_3 \neq i | X_1 = i] = p$, $i = 0, 1$ and $\Pr[X_3 \neq i | X_2 = i] = p$, $i = 0, 1$. Imposing this stronger symmetry conditions we get $b = 1/2$ and $a = \left(1 - \frac{3}{2}p\right)/(1-p)$ and (2.60)-(2.67) become

$$\Pr[X_1 = 0, X_2 = 0 | X_3 = 0] = 1 - \frac{3}{2}p \tag{2.68}$$

$$\Pr[X_1 = 0, X_2 = 1 | X_3 = 0] = \frac{p}{2} \tag{2.69}$$

$$\Pr[X_1 = 1, X_2 = 0 | X_3 = 0] = \frac{p}{2} \tag{2.70}$$

$$\Pr[X_1 = 1, X_2 = 1 | X_3 = 0] = \frac{p}{2} \tag{2.71}$$

$$\Pr[X_1 = 0, X_2 = 0 | X_3 = 1] = \frac{p}{2} \tag{2.72}$$

$$\Pr[X_1 = 0, X_2 = 1 | X_3 = 1] = \frac{p}{2} \tag{2.73}$$

$$\Pr[X_1 = 1, X_2 = 0 | X_3 = 1] = \frac{p}{2} \tag{2.74}$$

$$\Pr[X_1 = 1, X_2 = 1 | X_3 = 1] = 1 - \frac{3}{2}p \tag{2.75}$$

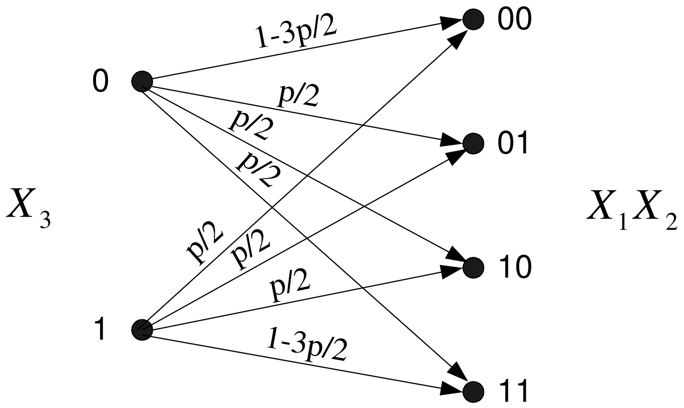

Fig. 20. Binary symmetric correlation model for Slepian-Wolf coding of three sources given in the form of an equivalent discrete memoryless channel.

We will consider this model to be the extension of the binary symmetric model assumed throughout the chapter to three sources. The equivalent discrete memoryless channel (DMC) which can be used to describe the correlation of the three sources, as given in equations (2.68-2.75) is shown in Fig. 20.

Assuming that X_1 is available at the decoder as side information, from the extension of the Slepian-Wolf theorem [8] to multiple sources [28, 33], the source coding rates R_2 and R_3 (in compressed bits per uncompressed bit) for the compression of X_2 and X_3, respectively, have to satisfy all the three following inequalities [28]

$$R_2 \geq H(X_2|X_1, X_3), \tag{2.76}$$

$$R_3 \geq H(X_3|X_1, X_2), \tag{2.77}$$

$$R_2 + R_3 \geq H(X_2, X_3|X_1). \tag{2.78}$$

Trying to minimize the rate R_3, i.e., bring it as close as possible to the theoretical limit $H(X_3|X_1, X_2)$, from (2.76) and (2.78), we get that the rate R_2 has to satisfy the inequality $R_2 \geq H(X_2|X_1)$. This corresponds to the corner point of the achievable region, i.e., to asymmetric Slepian-Wolf coding of more than two sources.

Expressing both limits in terms of p for the binary symmetric correlation model (stricter symmetry), we get

$$\begin{aligned} H(X_2|X_1) &= H(p) \equiv -p\,\log_2 p - (1-p)\,\log_2(1-p), \qquad (2.79)\\ H(X_3|X_1,X_2) &= p\,H\left(\frac{1}{2}\right) + (1-p)\,H\left(\frac{p/2}{1-p}\right)\\ &= 1 + (1-p)\,\log_2(1-p) - \left(1-\frac{3}{2}p\right)\,\log_2(2-3p)\\ &\quad -\frac{p}{2}\,\log_2 p. \qquad (2.80) \end{aligned}$$

We have plotted both (2.79) and (2.80), i.e., $H(X_2|X_1)$ and $H(X_3|X_1,X_2)$ respectively, as a function of p, in Fig. 21.

Using syndromes [7] or equivalently Wyner's scheme [16, 17], explained in the part on asymmetric Slepian-Wolf coding of two sources previously in this section, we can use the same approach here for the above setup of the three binary sources to get rates R_2 close to $H(X_2|X_1)$. Then this approach can be extended to get R_3 close to $H(X_3|X_1,X_2)$ and obtain the rate savings predicted by the theoretical limits of Fig. 21.

The rate savings are expected due to the improvement of the associated channel modelling the correlation between X_3 and the pair X_1, X_2, which is the DMC shown in Fig. 20, compared to the BSC. This rate savings have been confirmed through LDPC code designs matched to this DMC [34].

b. More Than Three Sources

The binary symmetric model discussed in the previous part can further be extended to more than three sources and it can also be extended to nonbinary sources, when

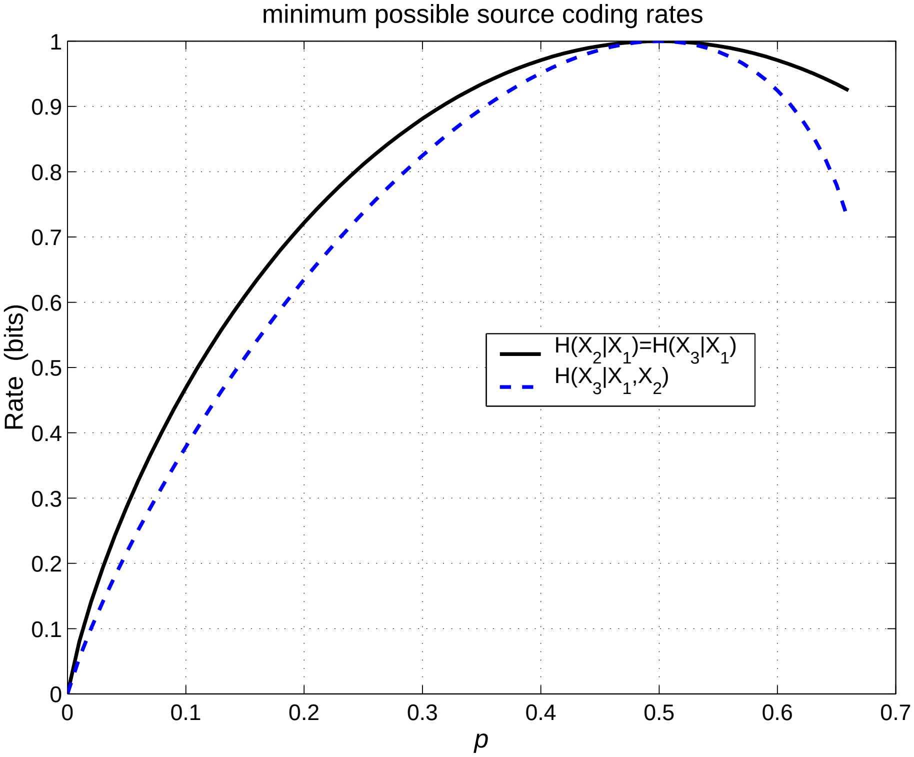

Fig. 21. $H(X_2|X_1)$ and $H(X_3|X_1, X_2)$ as a function of p: minimum achievable source coding rates.

each nonbinary symbol is represented with bits [35].

The only difference for more than three sources is that there are more than one parameters describing the statistical model. E.g., for four binary sources the binary symmetric model is described by p and a second parameter $0 \leq r \leq \frac{p}{2}$ [35]. In general as the number M of the binary uniform sources increases, the number of parameters describing the binary symmetric model, which results from the uniformity and from having BSC correlation between any two pairs of sources, i.e., $\Pr[X_j \neq i | X_k = i] = p,\ i = 0, 1,\ j \neq k,\ j, k = 1, 2, \ldots, M$, is also expected to increase.

c. Symmetric Coding

So far in this subsection we have not talked about symmetric Slepian-Wolf coding, as all the approaches focused on approaching the corner points. The problem with symmetric coding is that the way it was introduced in the previous subsection, it only applied to statistical models of sources with binary equiprobable outputs whose correlation is completely captured by their binary sum (BSC correlation).

This model is not practical for more than two sources, sometimes even for two sources it might not perform well enough. E.g., in the binary symmetric model with three sources, assuming that one of them, e.g., X_1, is available as side information at the decoder and trying to perform symmetric Slepian-Wolf coding of the other two sources X_2 and X_3 to approach the any point on the theoretical limit given by equations (2.76)-(2.78), it is not clear how the symmetric Slepian-Wolf coding approach for two sources in Subsection 2 should be modified to work well.

As long as in this simple setup symmetric Slepian-Wolf cannot be applied efficiently, which is still an open problem, it cannot be considered as efficient as the asymmetric coding approach, which can capture any correlation. Recently, other symmetric Slepian-Wolf approaches have been proposed independently of this work

[36, 37]. However, none of these approaches could match the elegance of using a single channel code as was done in the symmetric Slepian-Wolf coding approach for two sources in Subsection 2.

C. Asymmetric Slepian-Wolf Coding over Noisy Channel[†]

The Slepian-Wolf theorem [8] states that the lossless compression of the output of two correlated sources that do not communicate their outputs to each other, can be as efficient as if they communicated their outputs. This is true when their compressed outputs are jointly decompressed at a decoder. A system exploiting this property is shown in Fig. 1, where $\boldsymbol{Z} = [Z_1, Z_2, \ldots, Z_r]$ and $\boldsymbol{W} = [W_1, W_2, \ldots, W_m]$ are the compressed versions of the source output $\boldsymbol{X} = [X_1, X_2, \ldots, X_n]$ and $\boldsymbol{Y} = [Y_1, Y_2, \ldots, Y_n]$, respectively. $\boldsymbol{Z}$ and $\boldsymbol{W}$ are sent through channels to the joint decoder which yields $\hat{X}_i$ and $\hat{Y}_i$ with negligible probability of $\hat{X}_i \neq X_i$ and $\hat{Y}_i \neq Y_i$. An important assumption of the Slepian-Wolf theorem is that the channels through which $\boldsymbol{Z}$ and $\boldsymbol{W}$ are sent to the decoder are perfect. Practical schemes exploiting the potential of the Slepian-Wolf theorem were introduced in the previous section based on advanced channel codes.

Although LDPC codes seem to work better for the distributed compression problem, at least for binary sources, no similar approach has been proposed for distributed joint source-channel coding until now. Extra loss is introduced in this case due to the channel capacity of the nonideal channel(s). Only turbo codes have been used so far both for distributed joint source-channel coding of equiprobable memoryless correlated binary sources [3, 38] and the special case of source-channel coding of a single nonequiprobable memoryless binary source [39, 40, 41, 42]. Parallel to this

[†] ©2002 IEEE. Reprinted, with permission, from “Joint Source-Channel Coding of Binary Sources with Side Information at the Decoder Using IRA Codes”, Angelos D. Liveris, Zixiang Xiong, and Costas N. Georghiades, in *Proc. 2002 IEEE Workshop on Multimedia Signal Processing (MMSP ’02)*, Dec. 2002, pp. 53-56.

work and independently, another approach with concatenated low-density generator matrix (LDGM) codes was proposed [43, 44], which employs two LDPC codes per source instead of a single one.

Furthermore, due to the nonideal channel(s), the asymmetric case [3, 39, 40, 41, 42], i.e., joint source-channel coding with perfect side information at the decoder, is not equivalent to the symmetric case [38, 43, 44] (distributed joint source-channel coding). The asymmetric case employs one noisy channel, whereas the symmetric case two.

Here we focus on the asymmetric case, i.e., joint source-channel coding with side information at the decoder. It is not straightforward to apply LDPC codes in this case, not only due to their "conventional" channel encoding problems but also due to their design process which cannot take two different equivalent channels into account. On the other hand, systematic irregular repeat-accumulate (IRA) codes [30] seem to solve both problems. First, systematic IRA codes are a special form of LDPC codes without any "conventional" channel encoding problems, allowing similar design (optimization) procedure and suffering a very small loss in performance. Second, systematic IRA codes offer the potential of code design with different initialization for the systematic and parity part, which has already been used in [45].

In this section, we show how we can exploit these advantages when applying systematic and nonsystematic IRA codes to joint source-channel coding of an equiprobable memoryless binary source with side information at the decoder. The main idea is to view the system as transmitting the source over two channels. The first one is the actual channel through which the source-channel coded bits are sent to the decoder and describes the distortion experienced by the parity bits of the systematic IRA code. The second channel could be either a combination of the actual and the correlation channel (enhanced actual channel) or the correlation channel depending on whether the systematic part of the IRA code used is transmitted or not transmitted, respectively. This channel describes the distortion of the information bits. Taking the difference between the two channels into account in the design of the IRA codes, we obtain codes that perform better than turbo codes over the binary symmetric channel (BSC), the additive white Gaussian noise (AWGN) channel and the flat Rayleigh fading channel.

This section is organized as follows; Subsection 1 describes the overall system and introduces the equivalent way to view it through the two channels. In Subsection 2 the encoding, decoding and code design procedures for the IRA codes are explained and in Subsection 3 IRA codes are simulated and compared with turbo codes.

1. Problem Setup and the Two Channels

We consider the system of Fig. 1 with the following assumptions, which are used for the rest of this section:

- $\boldsymbol{X} = [X_1, X_2, \ldots, X_n]$, $\boldsymbol{Y} = [Y_1, Y_2, \ldots, Y_n]$, where the X_i's are i.i.d. equiprobable binary random variables and so are the Y_i's.

- X_i and Y_i are correlated with $\Pr[X_i \neq Y_i] = p < 0.5$.

- $\boldsymbol{Y}$ is available losslessly at the joint decoder and we try to compress $\boldsymbol{X}$ as efficiently as possible. Since the rate used for $\boldsymbol{Y}$ is its entropy $nR_Y = nH(Y_i) = n$ bits, the theoretical limit for lossless compression of $\boldsymbol{X}$ over an ideal channel is (from the Slepian-Wolf theorem [8]) $H(\boldsymbol{Z}) = r = nR_X \geq nH(X_i|Y_i) = nH(p) = n(-p\log_2 p - (1-p)\log_2(1-p))$, where $\boldsymbol{Z} = [Z_1, Z_2, \ldots, Z_r]$ is the source-channel encoder output when the input is $\boldsymbol{X}$. For the nonideal channel, the channel capacity C in bits per channel use, has to be accounted for, and so the limit becomes $r = nR_X \geq nH(p)/C$ [46], assuming BPSK modulation.

The first two assumptions are the same as those made at the beginning of this chapter (binary symmetric model). Imposing all the above three assumptions to the system of Fig. 1 we end up with the problem of *joint source-channel coding of an equiprobable memoryless binary source with side information* or asymmetric Slepian-Wolf coding over a noisy channel, shown in Fig. 2. A special case of this problem is when the binary source is nonequiprobable with entropy $H(p) = H(X_i)$ and then the side information Y_i is set equal to the most probable value of X_i.

There is an equivalent way one can view the system of Fig. 2 to allow the use of systematic channel codes, i.e., codes with codewords $[\boldsymbol{X} \mid \boldsymbol{P}]$ including all the information bits $\boldsymbol{X}$ plus the parity bits (redundancy) $\boldsymbol{P}$. To use an equivalent model of the system, we consider source-channel encoders with input $\boldsymbol{X}$ and output either $\boldsymbol{Z} = \boldsymbol{P}$ or $\boldsymbol{Z} = [\boldsymbol{X} \mid \boldsymbol{P}]$ based on a systematic channel code with codewords of the form $[\boldsymbol{X} \mid \boldsymbol{P}]$. When $\boldsymbol{Z} = \boldsymbol{P}$, we have a *nonsystematic source-channel code (NSSCC)* and when $\boldsymbol{Z} = [\boldsymbol{X} \mid \boldsymbol{P}]$ a *systematic source-channel code (SSCC)*. In the systematic case, the source-channel code rates are limited to $R_X = r/n \geq 1$, which does not restrict the design when the channel is very noisy.

The correlation between X_i and Y_i can be modeled with a BSC as discussed in the previous sections; X_i will be the input to this BSC and Y_i its distorted output. For the NSSCC, only the parity bits $\boldsymbol{Z} = \boldsymbol{P}$ are transmitted through the actual channel. In this case, we can view the overall system as transmission of a single codeword over two different channels; the systematic part $\boldsymbol{X}$ goes through the binary symmetric correlation channel and the parity part $\boldsymbol{Z} = \boldsymbol{P}$ through the actual channel, as shown in Fig. 22. For the SSCC, the source-channel encoder output $\boldsymbol{Z} = [\boldsymbol{X} \mid \boldsymbol{P}]$ passes through the actual channel. So the systematic part $\boldsymbol{X}$ can be viewed as passing through an “enhanced” channel (combination of actual and correlation channel) and the parity bits $\boldsymbol{P}$ through the actual channel. For both SSCCs and NSSCCs, the

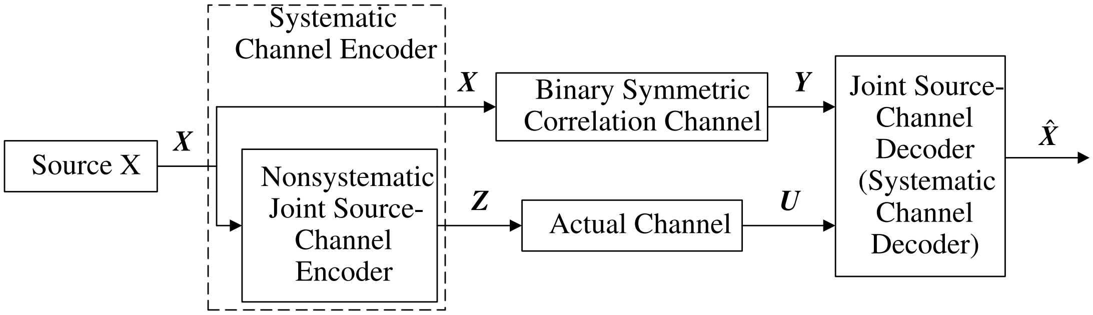

Fig. 22. The two equivalent channels for nonsystematic joint source-channel coding of $\boldsymbol{X}$ with side information $\boldsymbol{Y}$ at the decoder.

systematic irregular repeat-accumulate (IRA) codes are expected to perform very well, as they can be designed to have their systematic and parity part transmitted over two different channels.

2. Joint Source-Channel Coding with IRA Codes

Systematic IRA codes were introduced in [30] combining the advantages of LDPC codes (message-passing iterative decoding, code design, superior performance) and turbo codes (linear time encoding). They have a more constrained structure than LDPC codes and an ensemble of systematic IRA codes is described by the degree distribution polynomials $\lambda(x) = \sum_{i=2}^{J} \lambda_i x^{i-1}$ and $\rho(x) = x^{\alpha-1}$ [30]. A specific systematic IRA code is determined by its bipartite graph, which specifies the connections between the systematic bits and the check nodes.

a. Encoding

To jointly source-channel encode an information sequence of length n using an IRA code with degree distribution $\lambda(x)$ and $\rho(x) = x^{\alpha-1}$ and a given realization of the bipartite graph, we first determine the corresponding parity bit sequence $\boldsymbol{P}$ of length

$r = n/\left(\alpha \int_0^1 \lambda(x)dx\right)$ for an arbitrary input sequence $\boldsymbol{X}$ of length n. This is the output of the source-channel encoder sent through the channel if the source-channel code is nonsystematic (NSSCC). An encoding example for a NSSCC is given in Fig. 23. For a SSCC both $\boldsymbol{X}$ and $\boldsymbol{P}$ are transmitted through the actual channel.

b. Decoding

To explain the decoding algorithm we assume initially that the source-channel code is nonsystematic (NSSCC). The decoder tries to estimate $\boldsymbol{X}$ from $\boldsymbol{Y}$ and $\boldsymbol{U}$, the output of the actual channel when the input is $\boldsymbol{Z}$. The message-passing iterative decoding algorithm can be used where both the r parity nodes (corresponding to $\boldsymbol{U}$) and the n systematic nodes (corresponding to $\boldsymbol{Y}$) can be handled as variable (left) nodes [30]. The only difference is in the initialization of the algorithm with the log-likelihood ratios (LLRs) from the correlation or the actual channel. The LLR at the i^{th} systematic node is

$$q_{i,0}^{(sys)} = \log \frac{\Pr[x_i = 0|y_i]}{\Pr[x_i = 1|y_i]} = (1 - 2y_i) \log \frac{1-p}{p}, \tag{2.81}$$

$i = 1, 2, \ldots, n$, due to the binary symmetric correlation channel. The LLR at the i^{th} parity node is [47]

$$\begin{aligned} q_{i,0}^{(par)} &= \log \frac{\Pr[z_i = 0|u_i]}{\Pr[z_i = 1|u_i]} \\ &= \begin{cases} (1-2u_i)\log \frac{1-q}{q}, & \text{BSC} \\ 4\frac{\sqrt{E_s}}{N_0}u_i, & \text{AWGN} \\ 4\gamma_i \frac{\sqrt{E_s}}{N_0}u_i, & \text{Rayleigh fading} \end{cases}, \end{aligned} \tag{2.82}$$

$i = 1, 2, \ldots, r$, where for the BSC q is the crossover probability and for the additive white Gaussian noise (AWGN) and the flat independent Rayleigh fading channel, BPSK signaling is assumed, i.e., if $z_i = 0$ then $\sqrt{E_s}$ is transmitted and if $z_i = 1$

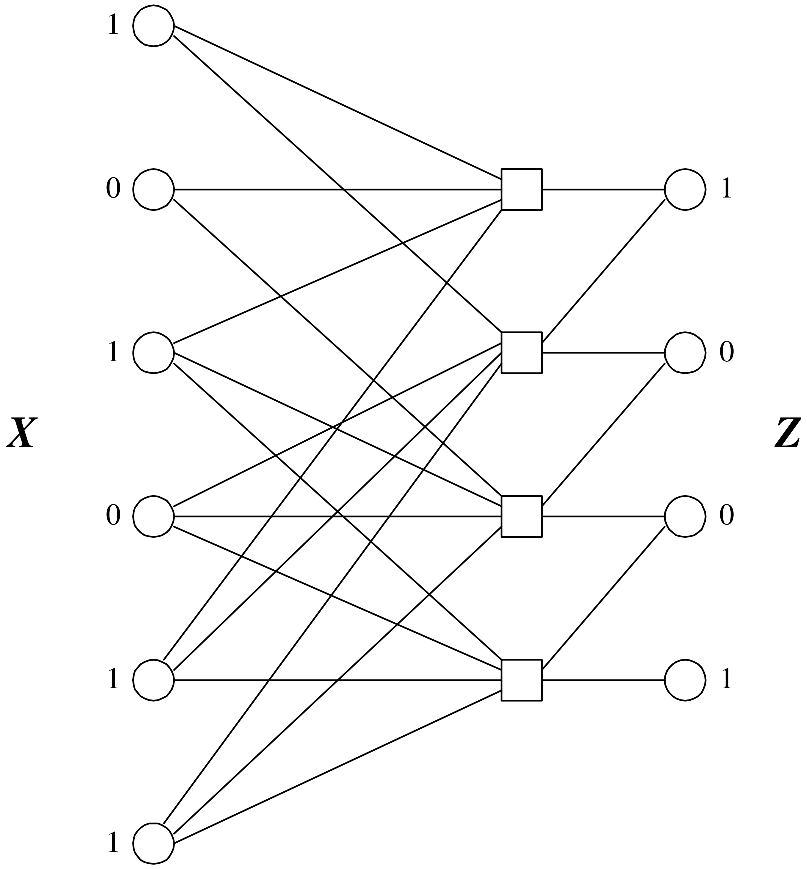

Fig. 23. Encoding with a NSSCC based on a systematic IRA code using the bipartite graph: at each check (square) node all the connected information nodes (cycles on the left) are added modulo 2 and the corresponding values of the parity nodes (cycles on the right) are determined. Here n is only 6, $\lambda(x) = 0.25x + 0.75x^2$, $\rho(x) = x^3$ and the rate of the associated systematic IRA code is 0.6, yielding a compression ratio of $3 : 2 = 1.5$ for the NSSCC.

then $-\sqrt{E_s}$ is transmitted. Furthermore, for the fading channel, perfect channel knowledge is assumed and so γ_i is a Rayleigh random variable with $\mathrm{E}[\gamma_i^2] = 1$, i.e., with probability density function $p_\Gamma(\gamma) = 2\gamma \mathrm{e}^{-\gamma^2}$. It is easy to see that in the SSCC, the parity nodes are initialized in the same way, but the information nodes are initialized by the sum of the LLRs of (2.81) and (2.82).

c. Code Design

The possibility of designing systematic IRA codes with different channel conditions for the systematic and the parity part is the main advantage of using IRA codes in joint source-channel coding with side information. We will follow the Gaussian approximation approach of [30]. So we start by defining the function $\phi(t)$ as

$$\phi(t) = \frac{1}{\sqrt{4\pi t}} \int_{-\infty}^{\infty} \tanh\left(\frac{a}{2}\right) \mathrm{e}^{-(a-t)^2/4t}\, da. \tag{2.83}$$

Assuming that the maximum allowable systematic node degree is J, the linear optimization of the systematic node degree distribution $\lambda(x) = \sum_{i=2}^{J} \lambda_i x^{i-1}$ for a given check node degree distribution $\rho(x) = x^{\alpha-1}$ is done by maximizing $\sum_{i=2}^{J} \frac{\lambda_i}{i}$ subject to the conditions [30]

$$\lambda(1) = 1, \tag{2.84}$$

$$F(x) > x, \quad \forall\, x \in [x_0, 1], \tag{2.85}$$

where $F(x)$ is defined as [30]

$$F(x) = \sum_{i=1}^{J} \lambda_i\, \phi\left(\mu_{\text{sys}} + (i-1)\, \phi^{-1}\left(\frac{\phi^2(f(x))}{x^{\alpha+1}}\right)\right), \tag{2.86}$$

and $x_0 = \phi(\mu_{\text{sys}})$. The function $f(x)$ is determined from the equation $\phi\left(f(x)\right) = x^{\alpha}\, \phi\left(\mu_{\text{par}} + f(x)\right)$ [30].

The only parameters we did not define in the above linear optimization procedure are the initial LLRs obtained from the channel for the systematic part μ_{sys} and the parity part μ_{par}. This procedure works well for Gaussian channels and initially [30] it was introduced for a single Gaussian channel where both parts of the systematic IRA codeword had the same channel LLR $\mu_{\text{sys}} = \mu_{\text{par}} = 4E_s/N_0$. The smallest value of E_s/N_0 that achieves a code rate R (R is an increasing function of $\sum_{i=1}^{J} \lambda_i/i$) is the *threshold* $(E_s/N_0)_{\text{th}} = R(E_b/N_0)_{\text{th}}$ of this ensemble of systematic IRA codes, i.e., the smallest value at which they can yield arbitrarily small bit error probability for large codeword length and large number of iterations.

In our case, we need to use different channel LLRs for the systematic part and the parity part, the former from the correlation channel and the latter from the actual channel. This is the property of systematic IRA codes we try to exploit in this joint source-channel coding scenario. Fixing the LLR corresponding to the correlation channel, we will try to minimize the threshold with respect to the actual channel. In addition, the above procedure works well when both channels are Gaussian. As all non-AWGN channel models used here are related to the AWGN channel (the BSC as an AWGN channel with quantized output and the flat independent Rayleigh fading channel as a time-varying AWGN channel), an approximate way to design the code, based on first mapping the channel parameters of one channel to the other [20] and then using the process outlined above, is expected to work reasonably well. Due to these approximations, to compare designs with different α, we evaluate the corresponding threshold using discretized density evolution [20].

We will consider three different correlation-actual channel combinations: BSC-BSC, BSC-AWGN and BSC-fading. For both the AWGN and fading actual channels the same codes will be used. So for the two different cases (BSC-BSC and BSC-AWGN/fading), we need to use a way of mapping the BSC threshold LLRs to the

AWGN ones. This is done based on the equality of the stability functions for the two channels [20]

$$\left(\frac{E_s}{N_0}\right)_{\mathrm{eq}} = -\log\left(2\sqrt{q(1-q)}\right), \tag{2.87}$$

where q is the BSC crossover probability and $(E_s/N_0)_{\mathrm{eq}}$ is the equivalent AWGN channel parameter. In [20] it was observed that this equality of the stability functions yielded reasonable threshold mappings between the channels for several regular and irregular LDPC codes. As for the enhanced channel (combination of correlation and actual channels for SSCCs), the equality of the stability functions yields

$$\left(\frac{E_s}{N_0}\right)_{\mathrm{eq}} = \frac{E_s}{N_0} - \log\left(2\sqrt{q(1-q)}\right), \tag{2.88}$$

where E_s/N_0 is the parameter of the actual AWGN channel.

3. Simulation Results

a. Binary Symmetric Channel

We first designed codes when the actual channel is a BSC, in order to compare with results in [3]. The BSC crossover probability is $q = 0.03$ and based on the method discussed in Subsection 2, the following rate 2/3 systematic IRA code resulted from the design process

$$\begin{aligned}\lambda(x) &= 0.27847x^2 + 0.25695x^{11} + 0.02840x^{12}\\ &\quad + 0.02454x^{47} + 0.10160x^{48} + 0.31004x^{49}, \end{aligned}\tag{2.89}$$

$$\rho(x) = x^{15}. \tag{2.90}$$

In the design, here and for the other channels, we did not allow any degree 2 information nodes and considered a maximum variable degree of 50. The corresponding NSSCC of (2.89) and (2.90) achieves 2 : 1 compression. The theoretical limit for this

Table IV. Gap of $E_b/N_0 = R_X E_s/N_0$ from Theoretical Limit in dB at BER$= 10^{-5}$ for the AWGN and Rayleigh Fading Channels

$R_X = 2$	AWGN				Rayleigh		
p	Turbo SSCC [40]	Turbo NSSCC [41]	IRA SSCC	IRA NSSCC	Turbo SSCC [40]	Turbo NSSCC [42]	IRA NSSCC
0.3	0.87	-	0.77	0.85	1.16	-	0.95
0.2	1.56	0.87	1.36	0.83	1.88	1.11	1.36
0.1	2.61	1.05	2.37	0.78	2.99	1.15	1.10

code rate $R_X = 0.5$ over the BSC with $q = 0.03$ is $H(X_i|Y_i) = H(p) \leq R_X \cdot C_{\text{BSC}} = R_X(1 - H(q)) = 0.403\,\text{bits}$. Using a length $n = 10^5$ for $\boldsymbol{X}$, we were able to get $\Pr[\hat{X}_i \neq X_i] < 10^{-6}$ at about $H(p) = 0.351\,\text{bits}$ (100 iterations in the decoder), which is closer to the limit than the $H(p) \approx 0.29$ of the turbo code design of [3]. To get this result 10^8 bits were simulated, i.e., 1000 blocks. More accurate optimization for the BSC is expected to increase the gains over turbo codes.

b. Additive White Gaussian Noise Channel

Our source-channel codes could outperform or come very close to the optimized turbo codes in [40, 41, 42] as shown in Table IV for the AWGN channel. In [40, 41, 42] the notation $R_c = 1/R_X$ is used for the channel code rate. We used up to 200 iterations in the decoder and an information sequence length of $n = 10^5$. The results refer to information bit error rate of less than or equal to 10^{-5} for 1000 simulated frames (10^8 simulated bits). The IRA codes used are given in Table V.

SSCCs exhibit the same behavior as the turbo codes of [40], i.e., the smaller the correlation between the source and the side information, the closer they get to the theoretical limit. Our SSCCs have slightly improved performance compared to the

Table V. IRA Codes Used for the $R_X = 2$ Results of Table IV

	$R_X = 2$					
	SSCC			NSSCC		
p	0.3	0.2	0.1	0.3	0.2	0.1
α	8	9	11	3	5	7
λ_3	0.25223	0.21391	0.15969	0.40416	0.19759	0.11892
λ_{10}		0.00033	0.04857			
λ_{11}	0.09645	0.04624	0.05405		0.00483	0.14062
λ_{12}	0.25906	0.29977	0.02768		0.28033	0.06640
λ_{13}			0.18146			
λ_{14}				0.38910		
λ_{31}	0.01889					
λ_{32}	0.05029					
λ_{33}	0.04113	0.02004				
λ_{34}	0.02192	0.02314	0.00472			
λ_{35}	0.02492	0.01978	0.01857			
λ_{36}	0.01086	0.01681	0.0333			
λ_{37}	0.00716	0.01704	0.02793			
λ_{38}		0.00564	0.02133			
λ_{39}		0.00131	0.00834			
λ_{40}			0.01152			
λ_{41}	0.21709		0.00355			
λ_{44}		0.33599				
λ_{49}			0.39929	0.06314		
λ_{50}				0.14360	0.51725	0.67406

SSCCs based on turbo codes. NSSCCs are more appropriate for higher correlation and seem to have a constant gap from the theoretical limit because of the IRA code structure imposed as well as the approximate IRA code design. This advantage of NSSCCs was independently observed in [41] and also lead to the improved results for turbo NSSCCs of Table IV.

c. Rayleigh Fading Channel

For the flat independent Rayleigh fading channel with perfect channel state information using the same NSSCCs as in Table V, we get the results of Table IV. There is again some improvement over those of [40] and about the same performance with [42]. The number of iterations in the decoder, the information sequence length and the number of simulation bits are the same as for the AWGN channel. For the SSCCs we do not use the ones designed for the AWGN channel, since the combining of the fading and BSC for an enhanced channel has to be done differently than the AWGN-BSC combination.

D. Binary Wyner-Ziv Coding[†]

In the previous sections of this chapter we considered several binary Slepian-Wolf coding schemes which perform close to the corresponding theoretical limits. We focused more on asymmetric Slepian-Wolf coding, for which an equivalent channel coding problem could be defined and good code designs from conventional channel coding could be employed to help us approach the Slepian-Wolf limit.

The link between asymmetric Slepian-Wolf coding and conventional channel coding was established based on Wyner's scheme [16, 17], i.e., using syndromes. Wyner's scheme suggested modelling the correlation between the source output and the side information with a channel and using syndromes [16, 17] to convert this source coding problem to an equivalent channel coding problem. Through this link between the two problems, whatever effort was made to come close to capacity in conventional linear

[†] ©2003 IEEE. Reprinted, with permission, from "Nested convolutional/turbo codes for the binary Wyner-Ziv problem", Angelos D. Liveris, Zixiang Xiong, and Costas N. Georghiades, in *Proc. 2003 IEEE International Conference on Image Processing (ICIP '03)*, Sept. 2003, pp. 601-604.

channel coding could be exploited in lossless *source coding with side information at the decoder* (SCSID) or asymmetric Slepian-Wolf coding to approach the Slepian-Wolf limit with the same closeness.

A more general approach to lossless SCSID (asymmetric Slepian-Wolf coding) is lossy SCSID, for which the theoretical limit was given by Wyner and Ziv [26] and thus, it will also be referred to as Wyner-Ziv coding. We are going to focus on the binary case, which was treated as an example in [26]. For binary lossy SCSID (binary Wyner-Ziv coding), it turns out that theoretically there is some small loss compared to the case in which the side information is available at the encoder as well. A theoretical solution for this binary problem, was recently given by Shamai, Verdú and Zamir [17, 48], as an extension/generalization of Wyner's scheme to the lossy case. Shamai, Verdú and Zamir suggested using nested linear codes, where a subcode of a good linear source code is a good linear channel code [17, 48].

However, to our knowledge, apart from the practical nested schemes in [27, 49], which refer to continuous Gaussian sources, there have not been any other practical designs for nested codes. Introducing such a novel turbo scheme by building and improving upon [27, 49] and the asymmetric Slepian-Wolf codes based on concatenated codes that were presented earlier in this chapter is the main contribution of this section.

In [27, 49] it has already been shown how the Shamai-Verdú-Zamir scheme can lead to nested convolutional codes. All the uncompressed bits are encoded by the source code, which is a convolutional code, using the Viterbi algorithm. Encoding means that the uncompressed bits are "quantized" to the nearest convolutional codeword in Hamming distance sense [50]. Using the index of this codeword, syndrome bits are generated following the asymmetric Slepian-Wolf coding approach. The syndrome bits are then used together with the side information bits at the decoder to recover the “quantized” bits.

The approach can be extended to concatenated convolutional codes to make the channel subcode stronger as already suggested in [49]. However, the design in [49] when operating close to the Wyner-Ziv limit encountered the following difficulty: when trying to further improve the source code, the channel code became worse. Here we show how to avoid this difficulty by combining the approaches in [27, 49] based on the previous section of this chapter on asymmetric Slepian-Wolf coding with concatenated codes.

1. System Setup and the Shamai-Verdú-Zamir Scheme

We consider the system of Fig. 6 with the binary symmetric assumptions, which have been used throughout this chapter. We repeat these assumptions here for convenience, where because of using convolutional codes later on in this section, we use capital N to denote the source output blocklength and the side information blocklength:

- $\boldsymbol{X} = [X_1, X_2, \ldots, X_N]$, $\boldsymbol{Y} = [Y_1, Y_2, \ldots, Y_N]$ and the X_i's are i.i.d. equiprobable binary random variables and so are the Y_i's.
- X_i and Y_i are correlated with $\Pr[X_i \neq Y_i] = p < 0.5$.

The additional assumptions we make in this section is that $\boldsymbol{Y}$ is available losslessly at the joint decoder (side information) and we try to compress $\boldsymbol{X}$ as efficiently as possible allowing some distortion $d = \Pr[\hat{X}_i \neq X_i]$. The theoretical limit for lossy compression of $\boldsymbol{X}$ is [26] $nR = H(\boldsymbol{Z}) \geq nR_{\mathrm{WZ}}(d) = ng^*(d)$ where $g^*(d)$ is given from [17, 26]

$$g^*(d) = l.c.e. \left\{ H(p * d) - H(d), (p, 0) \right\}, \tag{2.91}$$

for $0 \leq d \leq p$, where $H(u) = -u \log_2 u - (1-u) \log_2 (1-u)$, $u * v = u(1-v) + v(1-u)$ and $l.c.e.$ stands for the lower convex envelope of the function $H(p * d) - H(d)$ and the point $(p, 0)$. A plot of $R_{\mathrm{WZ}} = g^*(d)$ together with the rate-distortion function for joint encoding and decoding ($\boldsymbol{Y}$ available at both encoder and decoder), which is

$$R_{X|Y}(d) = \begin{cases} H(p) - H(d), & 0 \leq d \leq p \\ 0, & d \geq p \end{cases}, \tag{2.92}$$

is given in Fig. 24 for $p = 0.1$ and $p = 0.3$. In Fig. 24 the small loss because of not having the side information at the encoder results in the gap between the two curves.

The system of Fig. 6 with the above assumptions defines the problem of lossy *source coding with side information at the decoder* (SCSID) for a binary equiprobable memoryless source. This problem is also called the binary Wyner-Ziv coding problem.

In the lossless case $d \approx 0$ (practically $d \leq 10^{-5}$), an equivalent way to view the system of Fig. 6 in order to allow the use of channel codes is by modelling the correlation between X_i and Y_i with a binary symmetric channel (BSC) with crossover probability p. X_i will be the input to the channel and Y_i its output and the compressed version of $\boldsymbol{X}$, i.e., $\boldsymbol{Z}$, can be used to make $\boldsymbol{X}$ look like a codeword of a channel code.

Using a linear (N, K) binary block code in this binary case, there are 2^{N-K} distinct syndromes, each indexing a set of 2^K binary words of length N. We call the linear block code (all-zeros syndrome set) the original code. All the 2^{N-K} sets are disjoint and in each set the Hamming distance properties of the original code are preserved, i.e., all codes have the same performance over the binary symmetric correlation channel. In compressing, a sequence of N input bits is mapped into its corresponding $(N-K)$ syndrome bits. Thus, the compression ratio achieved with this scheme is $N : (N-K)$. This approach, known as *Wyner's scheme* [16, 17] for some time, has been used in [27] for the design of simple codes and in the previous sections for the design of turbo and LDPC codes.

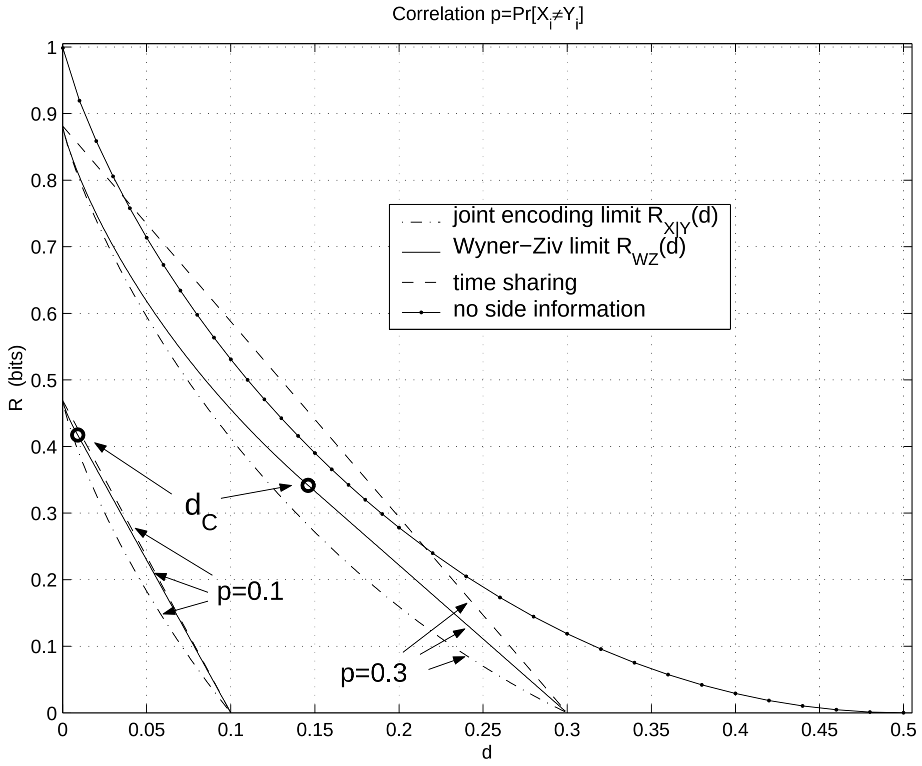

Fig. 24. $R_{\mathrm{WZ}}(d) = g^*(d)$ and $R_{X|Y}(d)$ as a function of $d = \Pr[\hat{X}_i \neq X_i]$. The limit for no correlation $1 - H(d)$ (no side information) is also plotted as well as the time sharing lines between the zero-distortion (Slepian-Wolf) and the zero-rate points. d_C is the largest d for which $g^*(d) = H(p * d) - H(d)$, i.e., for $d > d_C$, $g^*(d) < H(p * d) - H(d)$.

From Fig. 24 we can see that Wyner's scheme allows us to come close to the zero-distortion theoretical limits (points along the vertical axis of Fig. 24). Furthermore, with simple time sharing between the zero-distortion points (vertical axis) and the zero-rate points (horizontal axis), Wyner's scheme suffices to approach the intermediate theoretical limits (nonzero-distortion and nonzero-rate) for the case of high correlation ($p \leq 0.1$). However, for low correlation, such as $p = 0.3$ in Fig. 24, this time sharing can be even worse than the case of independent coding of $\boldsymbol{X}$ (not using the side information), so Wyner's scheme (for Slepian-Wolf coding) with time sharing can no longer bring us close to the limits.

To approach the Wyner-Ziv limit in this case, Shamai, Verdú and Zamir generalized Wyner's scheme using nested linear binary block codes [17, 48]. According to this nested scheme, a linear (N, K_2) binary block code is again used to partition the space of all binary words of length N into 2^{N-K_2} sets of 2^{K_2} elements, each indexed by a unique syndrome value. Out of these 2^{N-K_2} sets only $2^{K_1-K_2}$ $(K_1 \geq K_2)$ are used and the elements of the remaining $2^{N-K_2} - 2^{K_1-K_2}$ sets are "quantized" to the closest, in Hamming distance sense, binary word of the allowable $2^{K_1-K_2} \times 2^{K_2} = 2^{K_1}$ ones. This "quantization" can be viewed as a linear (N, K_1) binary block source code. Then the linear (N, K_2) binary block code can be considered to be a channel code which is a subcode of this source code (nesting).

To come close to the Wyner-Ziv limit with the Shamai-Verdú-Zamir scheme, both codes in the nested scheme have to be good, i.e., a good source code is needed with a good channel subcode [17, 48]. Knowing how to employ good channel codes based on Wyner's scheme $(K_1 = N)$ from the previous sections, we propose a scheme based on concatenated codes, where from the constructions based on syndromes the use of good channel codes is guaranteed. As for the source code, its operation resembles the operation of trellis coded quantization (TCQ) [50] and hence, it is expected to be a good source code.

Another remark about the Shamai-Verdú-Zamir scheme is that it can approach the theoretical limit when $d \leq d_C$ in Fig. 24 [26], since for this range $R_{\mathrm{WZ}}(d) = H(p * d) - H(d)$. Thus, a good source code of rate $R_S = \frac{K_1}{N}$ can be used to approach the limit $1 - H(d)$ $(R_S \geq 1 - H(d))$ and a good channel code of rate $R_C = \frac{K_2}{N}$ to approach the limit $1 - H(p * d)$ $(R_C \leq 1 - H(p * d))$, resulting in a Wyner-Ziv code rate of $R = \frac{K_1 - K_2}{N} = R_S - R_C \geq H(p * d) - H(d)$. From Fig. 24 this means that the

Shamai-Verdú-Zamir scheme works better for higher rates, e.g., with $R \geq 0.342\,\mathrm{bit}$ for $p = 0.3$, while for lower rates probably the best approach is time sharing between a good higher-rate nested code and the zero-rate point.

2. Nested Convolutional/Turbo Codes

The main thing that the extended Wyner's scheme made very clear is that the parity-check equations imposed by the binary quantizer should be complemented by the syndrome former so that the overall set of parity-check equations, i.e., those imposed by the quantizer together with those of the selected coset code, define a powerful binary channel code.

However, the theoretical description of this scheme does not provide enough details about its practical implementation. Since no other practical coding scheme has been proposed before for binary Wyner-Ziv coding, we look into several code possibilities in the following subsections. For instance, one thing that was not clear enough was whether *all* the source output bits have to be quantized, i.e., participate in the source code parity-check equations, or whether quantizing only a part of them suffices.

In the following we assume that we have a source output realization represented by a bit sequence $\boldsymbol{x}$ of length N and we discuss the encoding and decoding procedures for each practical scheme. We also try to analyze each scheme in an effort to identify its possible advantages or disadvantages.

a. Parallel Concatenation

A parallel concatenated Wyner-Ziv coding scheme is based on the assumption that not all the source output bits are quantized, as one of the component code is used for quantization of the source bits and the other component code is used just as a

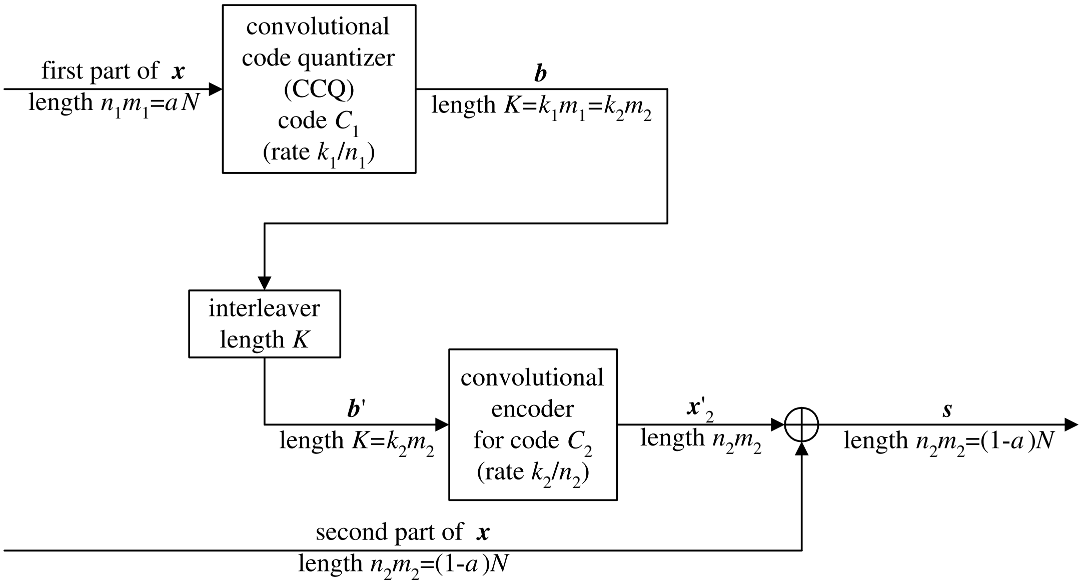

Fig. 25. The first parallel concatenated scheme ("parallel 1") analyzed for binary Wyner-Ziv coding; it looks very similar to the syndrome former for parallel concatenated codes (Fig. 17) with the difference that the first component code does not generate any syndrome bits. Instead, the first component code forces the syndromes of the first part of the source output sequence $\boldsymbol{x}$ to one of the codewords of the linear convolutional code C_1 through quantization.

syndrome former, as shown in Fig. 25. This assumption is a result of the syndrome forming approach we followed for parallel concatenated codes in the previous section, i.e., that the information sequence can be determined by one component code and used by the other component code to form the rest of the syndrome bits.

So, part of the bits in the sequence $\boldsymbol{x}$ of length N are quantized by a convolutional code quantizer (CCQ) [50], say $\alpha\,\%$. We denote the CCQ with C_1 and its code rate is assumed to be $\frac{k_1}{n_1}$. We further assume that $\alpha\,N = m_1\,n_1$, where m_1 is a positive integer. Then the resulting $m_1\,k_1$ index quantization bits $\boldsymbol{b}$ can be interleaved to form $\boldsymbol{b}'$.

$\boldsymbol{b}'$ is used as the information sequence into a second component code C_2 of rate $\frac{k_2}{n_2}$. Assuming that this component code C_2 is only used to to generate syndrome bits from the remaining (unquantized) $(1-\alpha)\,N = m_2\,n_2$ bits of the sequence $\boldsymbol{x}$. Following the syndrome forming approach of the previous section for parallel concatenated interleaved codes, we get $(1-\alpha)\,N = m_2\,n_2$ syndrome bits from this second component code, which are all the syndrome bits generated by this scheme.

It is not difficult to analytically derive the best possible performance of this code construction. To do so, we consider the rate-distortion bound for the code rate of the employed binary quantizer C_1 which is [9]

$$\frac{k_1}{n_1} \geq 1 - H\left(\tilde{d}\right) \tag{2.93}$$

where $H(\cdot)$ is the binary entropy function and

$$\tilde{d} = \Pr\left[X_i \neq \tilde{X}_i\right] = \frac{1}{\alpha\,N} \sum_{i=0}^{m_1\,n_1-1} |x_i - \tilde{x}_i| \tag{2.94}$$

is the average quantization over the $\alpha\,N$ bits of the sequence $\boldsymbol{x}$ that are quantized using the code C_1.

The code rate of the channel code that according to the extension of Wyner's scheme, should consist of all the source code parity-check equations as well as the parity-check equations used by the syndrome formers is in this case

$$\frac{m_1\,k_1}{N} = \frac{m_1\,k_1}{\alpha\,N}\,\alpha = \frac{m_1\,k_1}{m_1\,n_1}\,\alpha = \frac{k_1}{n_1}\,\alpha \tag{2.95}$$

This channel code rate is limited by the sum of the capacities of the "virtual" correlation channel between the $(1-\alpha)\,N$ unquantized source output bits $x_{\alpha\,N}, \ldots, x_{N-1}$ and the side information, which is a BSC with crossover probability p, and the "virtual" correlation channel between the $\alpha\,N$ quantized source output bits $x_0, x_1, \ldots, x_{\alpha\,N-1}$ and the side information, which is well approximated by two cascaded BSCs with crossover probability $\tilde{d}$ and p, respectively, or an equivalent BSC with crossover probability $\tilde{d} * p$. The BSC approximation of the quantization error is good enough if the CCQ C_1 operates close to the rate-distortion limit $1 - H\left(\tilde{d}\right)$ [17]. So the channel code rate $\frac{k_1}{n_1}\,\alpha$ can be bounded by

$$\frac{k_1}{n_1}\,\alpha \leq (1-\alpha)\;[1 - H(p)] + \alpha\;\left[1 - H\left(p * \tilde{d}\right)\right] \tag{2.96}$$

Solving this last inequality for $(1-\alpha)$, which is the Wyner-Ziv coding rate used, as $(1-\alpha)\,N$ syndrome bits are transmitted for N source output bits, we get

$$(1-\alpha) \geq \frac{\frac{k_1}{n_1} + H\left(p * \tilde{d}\right) - 1}{\frac{k_1}{n_1} + H\left(p * \tilde{d}\right) - H(p)} \geq \frac{H\left(p * \tilde{d}\right) - H\left(\tilde{d}\right)}{H\left(p * \tilde{d}\right) - H\left(\tilde{d}\right) + 1 - H(p)} \tag{2.97}$$

where in the last inequality we used equation (2.93).

Assuming that both the above inequalities can be made equalities, i.e., C_1 is a very good source code C_1 operating at the rate-distortion limit (second inequality) and C_1 and C_2 are combined to form a very good channel code operating almost at the capacity of the two “virtual” BSCs (first inequality),

$$(1-\alpha^*) = \frac{H\left(p * \tilde{d}\right) - H\left(\tilde{d}\right)}{H\left(p * \tilde{d}\right) - H\left(\tilde{d}\right) + 1 - H(p)} \tag{2.98}$$

is the lowest possible Wyner-Ziv coding rate that can be achieved by this parallel concatenated system. The distortion of this scheme at this rate is

$$d = \alpha^*\,\tilde{d} \tag{2.99}$$

The lowest possible Wyner-Ziv coding rate $(1-\alpha^*)$ for this first parallel concatenated scheme has been plotted in Fig. 26 as a function of the distortion d it achieves for $p = 0.27$, denoted as “parallel 1”. It is clear that this scheme is not good as it is equal to the Wyner-Ziv limit only for very small distortion d and for higher distortion

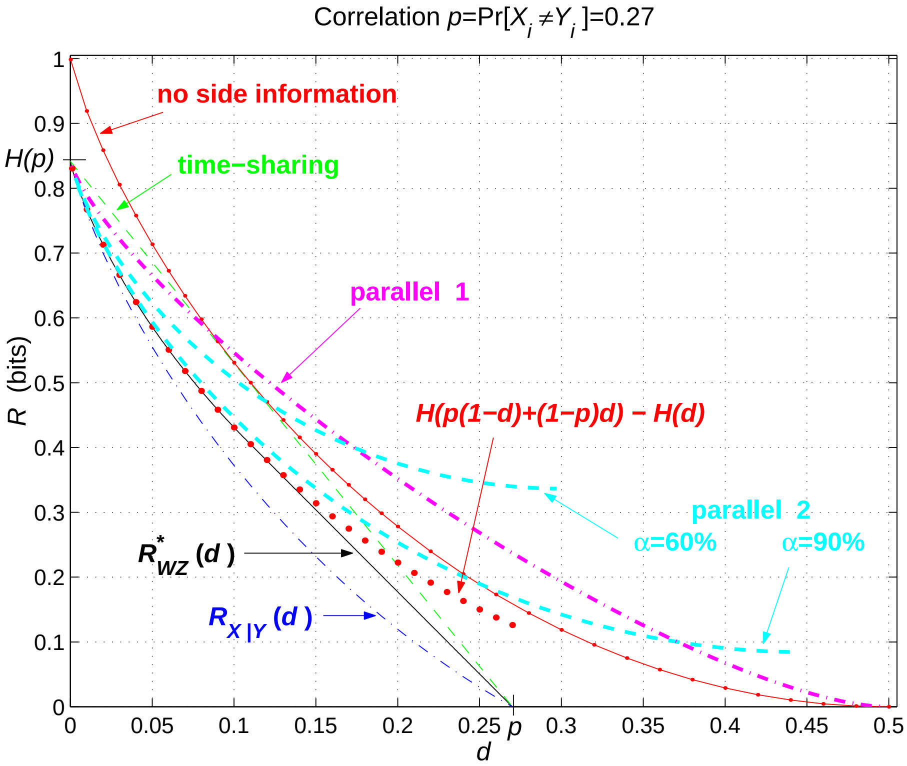

Fig. 26. The best achievable rate-distortion performance of the two parallel concatenated schemes compared with $R_{\mathrm{WZ}}(d)$ and $R_{X|Y}(d)$ as a function of d =Pr$\left[\hat{X}_i \neq X_i\right]$ for $p = 0.27$. The limit for no correlation $1 - H(d)$ (no side information) is also plotted as well as the time sharing lines between the zero-distortion (Slepian-Wolf) and the zero-rate points.

it is worse than time-sharing and worse than conventional binary quantization not utilizing the side information.

In this first scheme we assumed that each unquantized bit from the source output sequence $\boldsymbol{x}$ yields a syndrome bit and the "information sequence" $\boldsymbol{a}$ of length $m_1 k_1$ was determined only by the quantized source output bits. We can generalize further this parallel concatenated scheme by allowing it to have three component codes as shown in Fig. 27. The component codes C_1 and C_2 operate as before, doing quantization of $\alpha N = m_1 n_1$ source output bits and syndrome forming of $m_2 n_2$ source output bits given the information sequence $\boldsymbol{a}$ of length $m_2 k_2$, respectively, but the component code C_3 is now used to also determine part of the information sequence and form some additional syndrome bits.

In this more general parallel scheme, the interleaver length is $K = m_2 k_2 = (m_1 k_1 + m_3 k_3)$, the number of unquantized source output bits is $(m_2 n_2 + m_3 n_3) = (1-\alpha) N$ and number of syndrome bits formed is $m_2 n_2 + m_3 (n_3 - k_3) = (1-\alpha) N - m_3 k_3$. Working similarly as before, we have the same inequality (2.93) for the rate of the CCQ C_1 and for the rate of the overall parallel concatenated channel code, we have

$$\frac{m_2 k_2}{N} = \frac{m_1 k_1 + m_3 k_3}{N} = \frac{k_1}{n_1} \alpha + \frac{m_3 k_3}{N} \leq (1-\alpha) \left[1 - H(p)\right] + \alpha \left[1 - H\left(p * \tilde{d}\right)\right] \tag{2.100}$$

After some calculations, the last inequality can be rewritten as

$$\begin{aligned} \left(1 - \alpha - \frac{m_3 k_3}{N}\right) &\geq \frac{k_1}{n_1} \alpha + (1-\alpha) H(p) - \alpha \left[1 - H\left(p * \tilde{d}\right)\right] \\ &\geq H(p) + \alpha \left[H\left(p * \tilde{d}\right) - H(p) - H\left(\tilde{d}\right)\right] \end{aligned} \tag{2.101}$$

where in the last inequality we used equation (2.93).

Assuming that both the above inequalities can be made equalities, i.e., C_1 is a very good source code operating at the rate-distortion limit (second inequality) and C_1, C_2, and C_3 are combined to form a very good channel code operating almost at the capacity of the two "virtual" BSCs (first inequality),

$$\left(1 - \alpha - \frac{m_3 k_3}{N}\right)^* = H(p) + \alpha \left[H\left(p * \tilde{d}\right) - H(p) - H\left(\tilde{d}\right)\right] \tag{2.102}$$

is the lowest possible Wyner-Ziv coding rate that can be achieved by this parallel

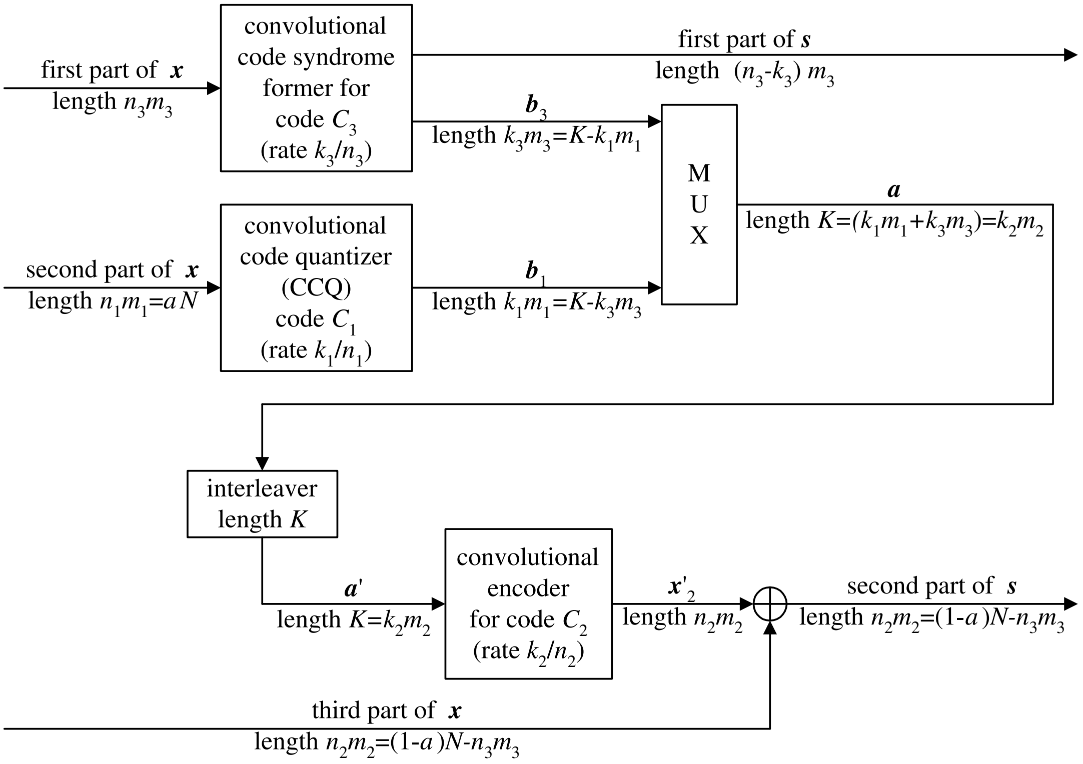
convolutional code syndrome former for code C_3 (rate k_3/n_3)
first part of $\boldsymbol{x}$
length n_3m_3
first part of $\boldsymbol{s}$
length $(n_3$-$k_3)$ m_3
$\boldsymbol{b}_3$
length k_3m_3=K-k_1m_1
M U X
$\boldsymbol{a}$
length K=$(k_1m_1$+$k_3m_3)$=k_2m_2
convolutional code quantizer (CCQ) code C_1 (rate k_1/n_1)
second part of $\boldsymbol{x}$
length n_1m_1=aN
$\boldsymbol{b}_1$
length k_1m_1=K-k_3m_3
interleaver length K
$\boldsymbol{a}'$
length K=k_2m_2
convolutional encoder for code C_2 (rate k_2/n_2)
$\boldsymbol{x}'_2$
length n_2m_2
second part of $\boldsymbol{s}$
length n_2m_2=(1-a)N-n_3m_3
third part of $\boldsymbol{x}$
length n_2m_2=(1-a)N-n_3m_3

Fig. 27. The second more general parallel concatenated scheme ("parallel 2") analyzed for binary Wyner-Ziv coding; it includes one more component code compared to the parallel concatenated scheme of Fig. 25. This extra component code determines part of the information sequence $\boldsymbol{a}$ and the other part is determined by the CCQ C_1.

concatenated system for a given α. The distortion of this scheme at this rate is

$$d = \alpha \, \tilde{d} \tag{2.103}$$

The lowest possible Wyner-Ziv coding rate $\left(1 - \alpha - \frac{m_3 \, k_3}{N}\right)^*$ for this second and more general parallel concatenated scheme has been plotted in Fig. 26 as a function of the distortion d it achieves for $p = 0.27$ with $\alpha = 60\%$ and $\alpha = 90\%$. It is denoted as "parallel 2". It is clear that this scheme is better than the first one ("parallel 1") but still not good enough as it is equal to the Wyner-Ziv limit only for small distortion d. Actually, the larger the percentage α of the quantized source output bits the smaller the loss of this scheme. However, only for $\alpha = 100\%$ is the theoretically expected loss equal to zero, as equation (2.102) also shows. For $\alpha = 100\%$ the only way to create a powerful channel code is through serial concatenated scheme, which will be considered next.

Before proceeding, we should note that the above analysis can be extended to cover cases where syndrome bits are also formed to modify the first component convolutional code C_1 that performs quantization. This modification might be needed to make the overall parallel concatenated scheme stronger as a channel code.

b. Serial Concatenation

The results of the previous subsection suggest that the whole binary source output sequence $\boldsymbol{x}$ has to be quantized and then syndromes can be formed based on the quantized version $\tilde{\boldsymbol{x}}$ of $\boldsymbol{x}$ or based on the quantization index $\boldsymbol{b}$.

The only concatenated structures that have been proposed before on the Gaussian Wyner-Ziv problem are [27, 49], which correspond to serially concatenated codes with and without an interleaver. Here we are going to combine and improve both these structures to achieve very good performance for the binary Wyner-Ziv coding

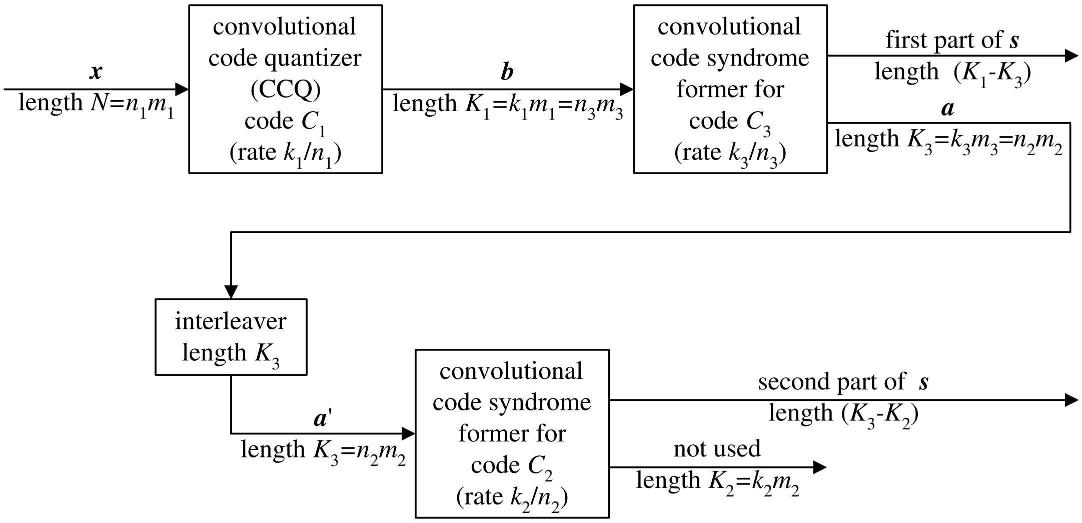
convolutional code quantizer (CCQ) code C_1 (rate k_1/n_1)
$\boldsymbol{x}$
length $N=n_1m_1$
$\boldsymbol{b}$
length $K_1=k_1m_1=n_3m_3$
convolutional code syndrome former for code C_3 (rate k_3/n_3)
first part of $\boldsymbol{s}$
length $(K_1$-$K_3)$
$\boldsymbol{a}$
length $K_3=k_3m_3=n_2m_2$
interleaver length K_3
$\boldsymbol{a}'$
length $K_3=n_2m_2$
convolutional code syndrome former for code C_2 (rate k_2/n_2)
second part of $\boldsymbol{s}$
length $(K_3$-$K_2)$
not used
length $K_2=k_2m_2$

Fig. 28. The encoder of the serially concatenated nested scheme consists of three component codes and an interleaver. The component code C_1 (rate $R_1 = \frac{k_1}{n_1}$) is a convolutional code quantizer and the other two codes C_2 (rate $R_2 = \frac{k_2}{n_2}$) and C_3 (rate $R_3 = \frac{k_3}{n_3}$) are used to generate syndromes.

problem.

Encoding: The general structure of the encoder of our scheme is shown in Fig. 28. The novelty of the scheme in Fig. 28 is the use of *three* component codes C_1, C_2 and C_3 as opposed to the two component codes used in [27, 49]. In [27] C_1 and C_3 were only used and in [49] only C_1 and C_2.

The encoding in Fig. 28 starts with the convolutional code C_1, which quantizes the uncompressed source output $\boldsymbol{x}$, which is a binary word of length N, to its nearest convolutional codeword in Hamming distance sense [50]. The corresponding index $\boldsymbol{b}$ of this convolutional codeword is the output of this convolutional code quantizer (CCQ) and is a binary sequence of length K_1.

The binary word $\boldsymbol{b}$ is then used to generate $(K_1 - K_3)$ syndrome bits (first part of $\boldsymbol{s}$) from the convolutional code C_3, which has rate $R_3 = \frac{k_3}{n_3} = \frac{K_3}{K_1}$. The convolutional code $R_{C,1}$ also generates K_3 information sequence bits $\boldsymbol{a}$. $\boldsymbol{a}$ is then interleaved and used to generate $(K_3 - K_2)$ more syndrome bits (second part of $\boldsymbol{s}$) with respect to the code C_2 with rate $R_2 = \frac{k_2}{n_2} = \frac{K_2}{K_3}$ code. So overall for N uncompressed bits $\boldsymbol{x}$, $(K_1 - K_2)$ syndrome bits $\boldsymbol{s}$ are generated, i.e., the Wyner-Ziv code rate is $R = \frac{K_1 - K_2}{N}$.

Decoding: The decoder structure of our nested scheme is shown in Fig. 29. The fact that the encoder has three component codes, whereas the decoder only two component decoders, is a result of joint decoding of the component convolutional codes C_1 and C_3. C_1 and C_3 are merged into a single convolutional code C_1' with rate $R_1' = \frac{K_3}{N}$ as shown in Fig. 29. Due to this merging of the two codes, taking into account the first part of the syndromes $\boldsymbol{s}$ in the C_1' decoder should be done as described in [27] for systematic trellis codes, which can be generalized to the nonsystematic convolutional codes used here.

Initially, the decoder for C_1' takes the length-N side information bit sequence $\boldsymbol{y}$ and the length-$(K_1 - K_3)$ syndrome bit sequence (first part of $\boldsymbol{s}$) as input. It produces two outputs, the length-N bit sequence $\hat{\boldsymbol{x}}$ (the estimate of the quantized version of $\boldsymbol{x}$) and extrinsic information for the length-K_3 bit sequence $\boldsymbol{a}$ [51]. This extrinsic information is interleaved and passed together with the $(K_3 - K_2)$ syndrome bits (second part of $\boldsymbol{s}$) to the decoder for C_2. The decoder for C_2 produces its extrinsic information for $\boldsymbol{a}$ which can be used as a third input to the decoder for C_1' and the decoding process can be continued iteratively.

Code Design: The main advantage of the above serially concatenated nested scheme is that by using three component codes, it allows greater flexibility in the code design than [27, 49]. There are two design guidelines. First, a good CCQ (code C_1) should be chosen from [50] that achieves a specific distortion d as close as possible to the rate distortion limit $1 - H(d)$. Then the codes C_2 and C_3 are chosen so that the concatenated scheme of C_1' and C_2 is a good channel code [51]. In our designs, given

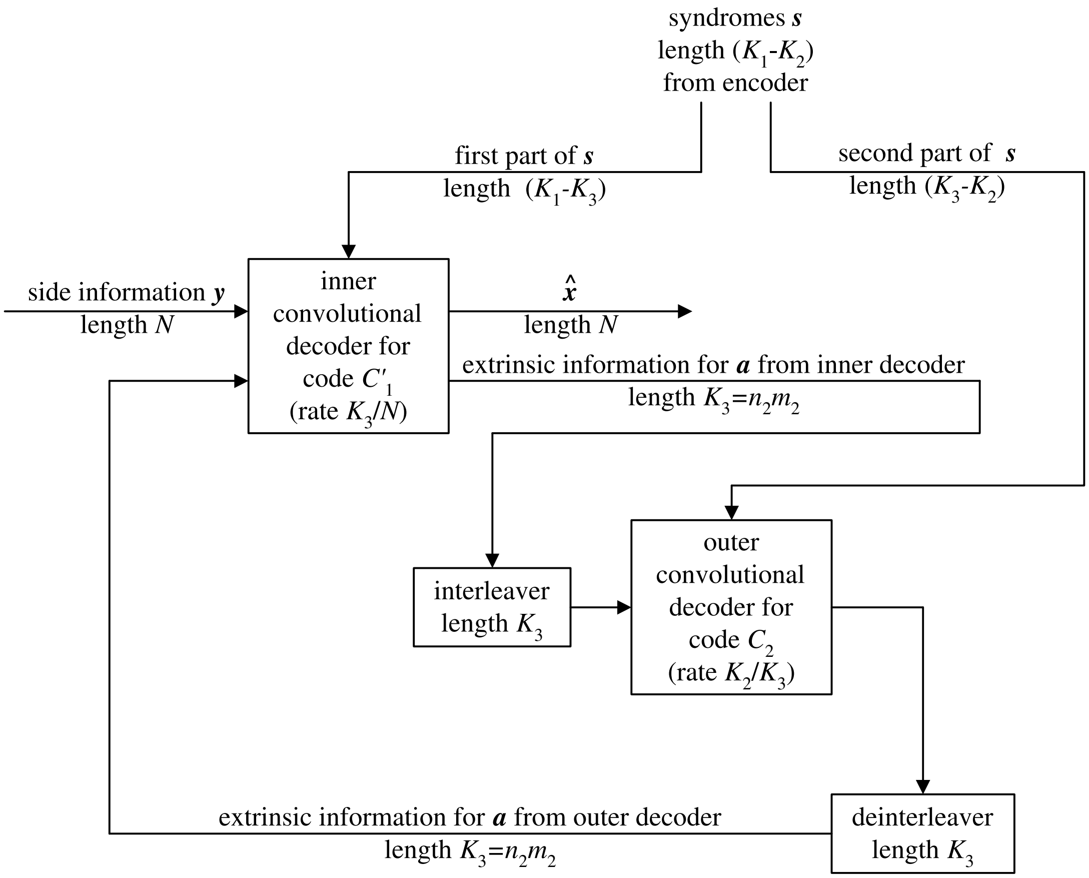
syndromes *s*
length (K_1-K_2)
from encoder
first part of *s*
length (K_1-K_3)
second part of *s*
length (K_3-K_2)
side information *y*
length *N*
inner convolutional decoder for code C'_1 (rate K_3/N)
$\hat{x}$
length *N*
extrinsic information for *a* from inner decoder
length K_3=n_2m_2
interleaver length K_3
outer convolutional decoder for code C_2 (rate K_2/K_3)
deinterleaver length K_3
extrinsic information for *a* from outer decoder
length K_3=n_2m_2

Fig. 29. The decoder of the turbo nested scheme consists of two component decoders. The component code C_1' of rate $R_1' = \frac{K_3}{N}$ is the convolutional code corresponding to the concatenation of the convolutional codes C_1 (rate $R_1 = \frac{K_1}{N}$) and C_3 (rate $R_3 = \frac{K_3}{K_1}$) and the code C_2 (rate $R_2 = \frac{K_2}{K_3}$) is the same as the one used at the encoder.

in the next subsection, the convolutional code C_1 has a large number of states, 16 or 32, and C_3 is a very simple convolutional code, which results in a lower complexity C_1', having only 4 states. This is an important requirement for good serially concatenated codes [51] and was causing problems in the channel coding performance in the design of [49]. The C_2 code is taken as a simple repetition code, which actually is one of the best choices for an outer code in a good serially concatenated code [51].

3. Simulation Results

For the simulation results we use $p = 0.27$ for the correlation, for which $d_C = 0.110$ and $R_{\text{WZ}}(d_C) = 0.405$. Therefore, we select the convolutional codes from TABLE VI in [50] that achieve distortion $d < 0.110$, i.e., codes C_1 with $R_1 = 2/3$ and $R_1 = 3/4$, and for a specific rate, we pick the code with the smallest d. So we end up with two CCQs, a 16-state $R_1 = 2/3$ and a 32-state $R_1 = 3/4$ [50]. The performance of these two CCQs in independent lossy source coding of an equiprobable binary source is given with the square points in Fig. 30.

For the simulation results of our nested scheme in Fig. 30, we took the $R_3 = 1/2$ 4-state $[1/(1+D^2),\ 1/(1+D^2)]$ code for $R_1 = 2/3$ and the $R_3 = 1/3$ 2-state $[1/(1+D),\ 1/(1+D),\ 1/(1+D)]$ code for $R_1 = 3/4$. The corresponding C_1' codes are the $R_1' = 1/3$ 4-state $[1,\ 1/(1+D^2),\ D/(1+D)]$ code and the $R_1' = 1/4$ 4-state $[(1+D+D^2)/(1+D),\ 1,\ (1+D+D^2)/(1+D),\ 1/(1+D)]$ code, respectively. The C_2 codes were repetition codes of rates $R_2 = 1/4$ and $R_2 = 1/3$, respectively. Thus, the nested schemes with $R_1 = 2/3$ and $R_1 = 3/4$ used Wyner-Ziv coding rates of $R = 7/12$ and $R = 2/3$, respectively.

The simulation results of our nested turbo scheme are shown in Fig. 30, for $N = 3 \cdot 10^5$ and 50 iterations. The two Wyner-Ziv codes are only 0.083 and 0.085 bits away from the theoretical Wyner-Ziv limit, respectively, and outperform by 0.023

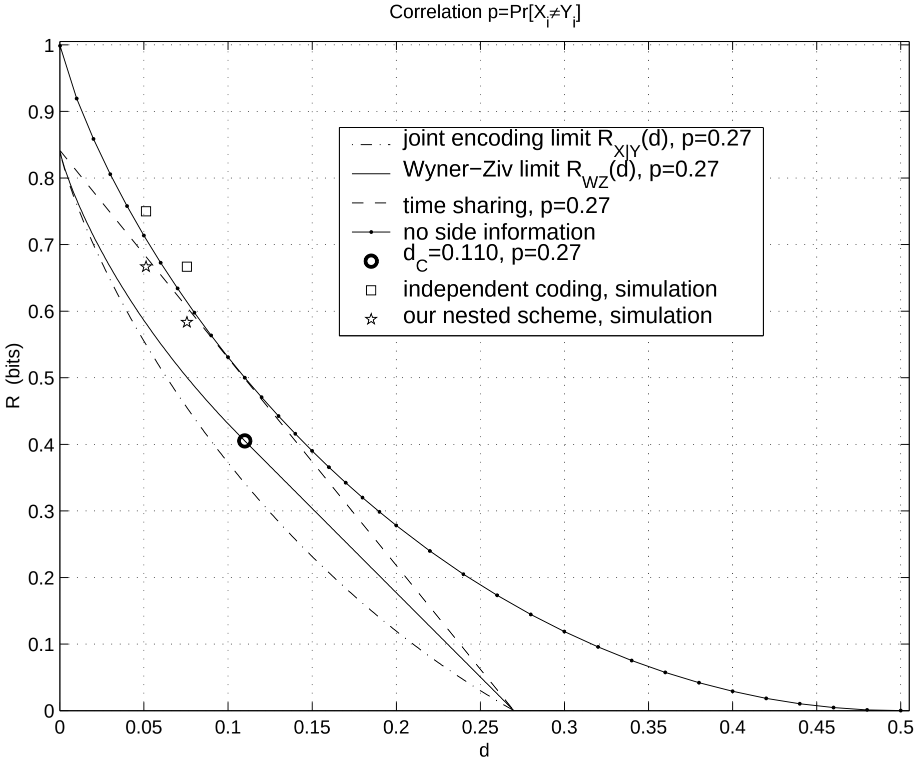

Fig. 30. The simulated performance of our nested scheme together with the simulated performance of the CCQ (rate R_1) used, for correlation $p = 0.27$. $R_{X|Y}(d)$, $R_{\mathrm{WZ}}(d)$, the time sharing line, the independent coding theoretical limit $1 - H(d)$ and d_C are also shown. For the simulation results of our nested scheme $N = 3 \cdot 10^5$.

and 0.015 bits, respectively, the time sharing scheme (between Slepian-Wolf and the zero-rate point).

E. Discussion and Conclusions

Throughout this chapter we proposed several distributed coding schemes focusing more on the coding with side information at the decoder problems. The connection of all the presented schemes with classical channel coding and source-channel coding problems was the key idea that led to the improved performance of the proposed schemes when compared to their competing counterparts which were missing the classical coding analog, especially in the channel coding component.

Our model was limited to the binary symmetric case but this forms the basis for the extension to nonsymmetric binary scenarios [52, 53] and to nonbinary scenarios, which have to be used when quantizing continuous-valued sources, e.g., Gaussian [3, 27, 49, 54, 55, 56], in Wyner-Ziv problems.

The suggested coding schemes find very interesting applications in several classical and network communications problems, e.g., optimization of retransmissions [57] and distributed source-channel over multiple access channels [58]. Another promising application of distributed coding is Wyner-Ziv coding for video [59, 60, 61, 62].

CHAPTER III

QUANTIZATION OF CODED CHANNEL MEASUREMENTS

In this chapter, we study the effects of quantization at the output of an additive white Gaussian noise (AWGN) channel on the achievable rates through the system and on the performance of several channel codes. We focus more on the binary input AWGN (BIAWGN) channel and on low-density parity-check (LDPC) codes. The channel decoder works with unlimited precision. The quantizer design is based on optimizing either the mean squared error (MSE) between the input and the output of the quantizer, or the equivalent channel cutoff rate or the equivalent channel mutual information. We start by considering uniform scalar quantizers and gradually, better quantization performance is considered with more complex quantizer designs. Nonuniform scalar quantizers are designed trying to approach the best theoretically achievable performance for given precision. All the approaches are compared for several channel codes in search of the most appropriate quantizer design criterion at given quantization complexity.

A. Introduction

The outstanding performance of turbo codes resulted in their use for practical applications, such as the third generation wireless communication systems. When applying them in such systems, more implementation issues need to be studied, with quantization being one of the most important among them. Significant work has already been done in this field considering quantization of either the received channel values alone [63, 64] or both the received channel values and the internal values in the decoder [65, 66, 67, 68, 69]. More recently, the reinvention of low-density parity-check (LDPC) codes with performance superior to that of turbo codes led to similar work on quantization for LDPC codes [70, 71].

Quantization of the received channel values or *channel measurements* has previously been studied for convolutional codes [72, 73] and for block codes [74]. This problem has also been studied in a more general context, irrespective of channel codes, by examining the loss on the achievable rate through the limited-precision communication system and other important fundamental quantities [75, 76, 77, 78, 79].

Another very related problem to the quantization of channel measurements is the "integer metrics" problem [80, 81], which imposes some additional practical constraints on the quantizer.

All this previous work led to the conclusion that the quantizers for the channel output values should not be designed conventionally, i.e., trying to minimize the mean squared error (MSE) between the input and the output of the quantizer. Instead, considering the optimization of other quantities, such as the equivalent channel cutoff rate and mutual information, yields better decoding performance. However, there are still a number of open questions regarding the effect of quantizing only the channel output values even with unlimited decoder precision. How meaningful would it be to introduce a more complex quantizer, i.e., are there any gains compared to simple uniform scalar quantization? How do more complex quantizer designs based on different criteria compare? Do cutoff rate maximization and mutual information maximization still exhibit a clear advantage over MSE minimization, even for more complex quantization schemes? How robust are all these quantizers, i.e., which of the criteria leads to the same quantizer being able to perform similarly over a range of SNRs? These questions are the motivation for this work.

We investigate the throughput loss in a binary input additive white Gaussian noise (BIAWGN) channel for given limited precision at its output. To get a better idea of the ultimate limits, we study the gains of complex quantization schemes.

Although such an approach might not look so meaningful, i.e., lowering the complexity by quantizing the channel output to a limited number of levels but increasing it with a sophisticated quantizer, it is justified from two different perspectives.

First, it is an interesting theoretical problem to come closer to the maximum possible achievable rates of a system with a single constraint, the limited precision at the channel output. Second, this maximum throughput will show whether there is anything to gain when replacing a simple uniform scalar quantizer with a nonuniform quantizer or an even more complex quantizer.

We design several uniform and nonuniform, MSE minimizing, cutoff rate maximizing and mutual information maximizing scalar quantizers and compare them in terms of MSE, capacity and cutoff rate. Then these quantizers are compared by simulating several LDPC codes and examining the effect of the quantizer choice on the code performance. The purpose of this work is to ascertain which criterion is the most appropriate for the design of channel output quantizers in the case of powerful soft-input decoding codes. Furthermore, we find it very interesting to examine whether a more complex quantizer results in better (lower) bit error rate (BER) at the output of the channel decoder and to study the robustness of each criterion for a wide SNR range.

The chapter is organized as follows. We start with the definition of the system model in Section B and the introduction of the three quantization criteria, the MSE, the cutoff rate and the mutual information. In Section C different uniform scalar quantizers are designed based on the three different criteria of mutual information maximization, cutoff rate maximization and MSE minimization and compared. Nonuniform scalar quantizers are then designed in Section D, based on the three different criteria, and compared not only between themselves but also to their uniform counterparts of Section C. Section E presents simulation results for the performance of LDPC codes with several of the quantizers presented in the previous sections. The conclusions sum up the chapter and discuss still open directions of this problem.

B. System Model

As already established in [63, 64, 66, 75, 77, 78, 79, 80], scalar quantization of the output of the BIAWGN channel with q bits, produces an equivalent channel with 2 inputs and $L = 2^q$ outputs, as shown in Fig. 31. We will assume that L takes any positive integer value greater than one, so q is not necessarily an integer.

For soft-input processing of the quantized outputs, the only parameter of interest in these L outputs is the probability with which each one of them occurs, unlike the conventional minimum mean squared error (MMSE) quantization where the quantizer levels are also important. We will follow the notation of [64] for the transition probabilities P_{ij}

$$P_{ij} \equiv \Pr\left[y_k \in T_j | x_k = i\right] = \int_{T_j} p(y_k|x_k = i)dy_k, \quad i = 0, 1, \quad j = 0, 1, \ldots, L-1, \tag{3.1}$$

where x_k and y_k are respectively the transmitted value and the unquantized received value at the time instant k, $T_j = (a_j, a_{j+1})$ is the j^{th} quantization interval, $j = 0, 1, \ldots, L-1$, with $a_0 = -\infty$ and $a_L = \infty$ and $p(y_k|x_k = i)$ is the conditional probability density function for the received unquantized value y_k, given that $x_k = i$, $i = 0, 1$ was transmitted.

For the BIAWGN channel with BPSK modulation, i.e., $y_k = 1 - 2x_k + n_k$ and

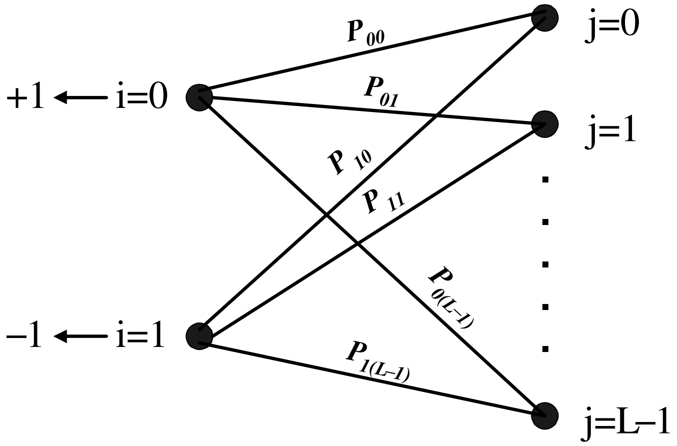

Fig. 31. The equivalent channel of a BIAWGN channel with $L = 2^q-$level quantized output. L can be any positive integer greater than one, i.e., q might not be an integer.

$n_k \sim N(0, \sigma^2)$, it is

$$P_{0j} = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{a_j}^{a_{j+1}} \mathrm{e}^{-\frac{(u-1)^2}{2\sigma^2}}\, du = \begin{cases} \mathrm{Q}\left(\frac{1-a_{j+1}}{\sigma}\right) - \mathrm{Q}\left(\frac{1-a_j}{\sigma}\right), & a_{j+1} \leq 1 \\ 1 - \mathrm{Q}\left(\frac{a_{j+1}-1}{\sigma}\right) - \mathrm{Q}\left(\frac{1-a_j}{\sigma}\right), & a_j \leq 1 \leq a_{j+1} \\ \mathrm{Q}\left(\frac{a_j-1}{\sigma}\right) - \mathrm{Q}\left(\frac{a_{j+1}-1}{\sigma}\right), & a_j \geq 1 \end{cases}, \tag{3.2}$$

where $\mathrm{Q}(t) = \frac{1}{\sqrt{2\pi}} \int_t^\infty \mathrm{e}^{-u^2/2}\, du$.

We should note that this BIAWGN model is introduced here for simplicity. However, we will sometimes also use the more practical model $y'_k = (1 - 2x_k)\sqrt{E_s} + n'_k$ with $n'_k \sim N\left(0, \frac{N_0}{2}\right)$, where changing the SNR means changing the transmitted energy and not varying the noise power. Whenever this latter more practical model is used, it will be explicitly stated and the affected quantities will be denoted with $'$, e.g., the noise sample is denoted with n'_k in the practical model, as opposed to n_k in the default model because it has different variance. For the same SNR $\frac{E_s}{N_0} = \frac{1}{2\sigma^2}$, the transition probabilities P_{ij} in both models are the same and if in the default model a set of boundaries $\{a_j\}$ is chosen, the equivalent set of boundaries in the practical model is $\{a'_j\} = \{a_j\sqrt{E_s}\}$.

Assuming equiprobable binary input to the BIAWGN channel and symmetric quantization around zero, we need to select the probability distribution of the $L = 2^q$ outputs of the quantizer. Due to the symmetry of the problem under these assumptions, $a_j = -a_{L-j}$, $j = 0, 1, \ldots, L$, and so we need to determine $\lceil L/2 \rceil - 1$ region boundaries $a_1, a_2, \ldots, a_{\lceil L/2 \rceil - 1}$ instead of $L - 1$, where $\lceil L/2 \rceil$ is the smallest integer that is greater than or equal to $L/2$, to account for both odd and even L.

The quantities that are modified by this choice are the MSE between the input and the output of the quantizer, the cutoff rate of the equivalent channel of Fig. 31 and the mutual information between the input and the output of the equivalent channel, i.e., the information rate that can pass through the limited precision system. In the next sections, we will try to determine the set of boundaries $\{a_j\}$ that optimizes each of these quantities. So, we will get a set of $\{a_j\}$ that minimizes the MSE, another set $\{a_j\}$ that maximizes the cutoff rate, and another set $\{a_j\}$ that maximizes the mutual information.

Before proceeding with the quantizer design we provide the expressions for each of these quantities in the following subsections.

1. Mean Squared Error

In the case of minimum mean squared error (MMSE) quantization, the quantizer is not only determined by its region boundaries a_j, $j = 1, 2, \ldots, L - 1$, but also by the values (levels) v_j, $j = 0, 1, \ldots, L - 1$ representing each region T_j. In general, the Lloyd-Max algorithm [82] should be used to optimize the quantization boundaries and levels. For a general L-level scalar quantizer, uniform or nonuniform, the quantization

MSE $D_{B,L}$ is

$$D_{B,L} = \sum_{j=0}^{L-1} \int_{a_j}^{a_{j+1}} (v_j - y_k)^2 \, p(y_k) \, dy_k \tag{3.3}$$

with $a_0 = -\infty$, $a_L = \infty$ and $p(y_k) = (p(y_k|x_k = 0) + p(y_k|x_k = 1))/2$. Substituting for $p(y_k)$ and using (3.1), (3.3) becomes

$$D_{B,L} = 1 + \sigma^2 - \sum_{j=0}^{L-1} v_j \, (c_{0j} + c_{1j}) + \frac{1}{2} \sum_{j=0}^{L-1} v_j^2 \, (P_{0j} + P_{1j}), \tag{3.4}$$

where $c_{ij} = \int_{a_j}^{a_{j+1}} y_k \, p(y_k|x_k = i) \, dy_k$, $i = 0, 1$, $j = 0, 1, \ldots, L-1$. Due to the quantizer symmetry, which means that $P_{0j} = P_{1(L-1-j)}$, $v_j = -v_{L-1-j}$ and $c_{0j} = -c_{1(L-1-j)}$, the last expression of $D_{B,L}$ can also be written as

$$D_{B,L} = 1 + \sigma^2 - 2 \sum_{j=0}^{L-1} v_j \, c_{0j} + \sum_{j=0}^{L-1} v_j^2 \, P_{0j} \tag{3.5}$$

After some computations we can also express c_{ij} as

$$c_{ij} = \begin{cases} (1-2i) \, P_{i0} - \frac{\sigma}{\sqrt{2\pi}} \, \mathrm{e}^{-\frac{(a_1 - 1 + 2i)^2}{2\sigma^2}} & \text{for } j = 0 \\ (1-2i) \, P_{ij} + \frac{\sigma}{\sqrt{2\pi}} \left(\mathrm{e}^{-\frac{(a_j - 1 + 2i)^2}{2\sigma^2}} - \mathrm{e}^{-\frac{(a_{j+1} - 1 + 2i)^2}{2\sigma^2}} \right) & \text{for } j = 1, 2, \ldots, L-2 \\ (1-2i) \, P_{i(L-1)} + \frac{\sigma}{\sqrt{2\pi}} \, \mathrm{e}^{-\frac{(a_{L-1} - 1 + 2i)^2}{2\sigma^2}} & \text{for } j = L-1 \end{cases} \tag{3.6}$$

Since this MSE minimization quantization problem has been well studied in a source coding perspective, we can approximately determine the minimum theoretically achievable MSE $D^*_{B,q}$ of a quantizer with rate q bits from rate distortion theory. Note that we use the quantizer rate q instead of the number of quantization levels L

to allow for the most general rate distortion function. The equality $q = \log_2 L$ holds only for scalar fixed-rate quantizers.

We are not aware of any exact expression of the distortion-rate function of this bimodal Gaussian distribution, but we can use the bounds of [9, p. 370, problem 8], according to which

$$\frac{1}{2\pi e}\, 2^{2\,(h(Y)-q)} \leq D^*_{B,q} \leq 2^{-2q}\,\sigma_Y^2 \tag{3.7}$$

where $Y = 1 - 2\,X + N$ is the output of the BIAWGN channel with input X and noise N, and $\sigma_Y^2 = 1 + \sigma^2$ is its variance, where σ^2 is the variance of the noise N. The differential entropy $h(Y)$ of Y can be determined numerically from

$$\begin{aligned} h(Y) &= -\int_{-\infty}^{\infty} \frac{p(y|x=0)+p(y|x=1)}{2} \log\left(\frac{p(y|x=0)+p(y|x=1)}{2}\right) dy \\ &= 1 + \frac{1}{2}\log_2\left(2\pi e \sigma^2\right) - \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-\frac{u^2}{2}} \log_2\left(1 + e^{-\frac{2}{\sigma^2}-\frac{2u}{\sigma}}\right) du \end{aligned} \tag{3.8}$$

Both the upper and lower bounds on $D^*_{B,q}$ have been plotted in Fig. 32, from where it is clear that for lower $\frac{E_s}{N_0} = \frac{1}{2\sigma^2}$ the two bounds coincide but not for higher SNRs. We will use both bounds as a reference to the MSE achieved by different quantizer designs in the next sections.

The MSE is the only quantity of the three to be optimized, i.e., the MSE, the cutoff rate and the mutual information, that is affected by the choice of the statistical model. This means that if we use the more practical model with E_s being the energy per transmitted coded bit and $\frac{N_0}{2}$ the noise variance, instead of 1 and $\sigma^2 = \frac{N_0}{2E_s}$, respectively, used to generate the curves in Fig. 32, all the curves in Fig. 32 have to be scaled by E_s, i.e., $D'^*_{B,q} = E_s\, D^*_{B,q}$. The results are shown in Fig. 33, where the more practical model has been used and the bounds on the minimum possible scaled MSE $D'^*_{B,q}/N_0$ have been plotted. Equivalently, one can think of Fig. 32 being scaled

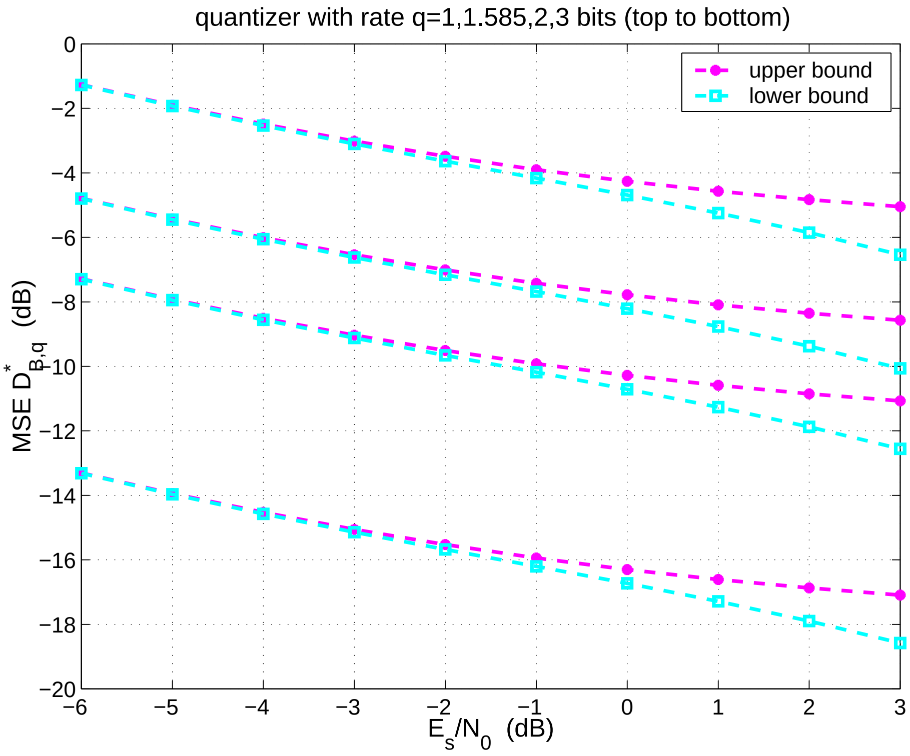

Fig. 32. The upper and lower bounds on the minimum achievable MSE $D^*_{B,q}$ of a rate q bits quantizer, as a function of E_s/N_0 for $q = 1, \log_2 3, 2, 3$ bits.

by $\frac{E_s}{N_0}$ to yield Fig. 33.

2. Cutoff Rate

From [63, 73, 75, 76] the cutoff-rate for the equivalent equiprobable binary input discrete memoryless channel of Fig. 31 is

$$R_{B,L} = 1 - \log_2 \left(1 + \sum_{j=0}^{L-1} \sqrt{P_{0j} P_{1j}} \right) \tag{3.9}$$

where the P_{ij} are defined in (3.1).

Unfortunately, there is no such reference as the cutoff rate for a given quantizer rate q, similar to the distortion-rate function of the previous subsection. So, the

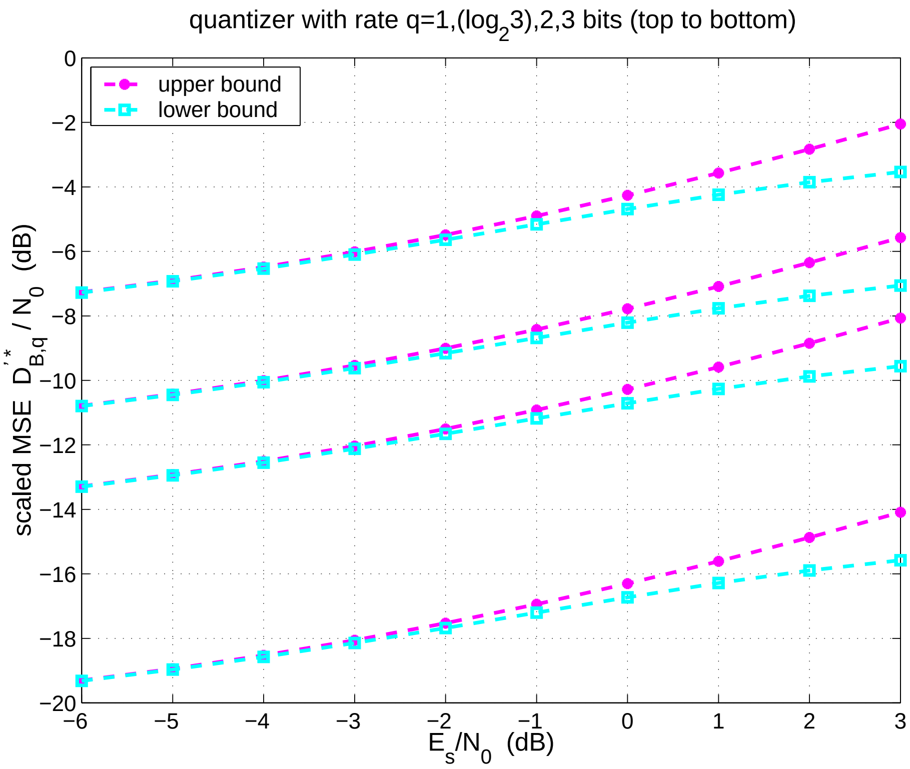

Fig. 33. The upper and lower bounds on the minimum achievable scaled MSE $D'^{*}_{B,q}/N_0$ of a rate q bits quantizer, as a function of E_s/N_0 for $q = 1, \log_2 3, 2, 3$ bits using the more practical model.

only reference we will be using to compare with the cutoff rate achieved by several quantizer designs is the cutoff rate of the unlimited precision BIAWGN channel $R_{B,\infty}$, which is

$$R_{B,\infty} = 1 - \log_2\left(1 + e^{-\frac{E_s}{N_0}}\right) \tag{3.10}$$

3. Mutual Information

The mutual information $I_{B,L}$ of the equivalent channel of Fig. 31 with equiprobable and independent binary inputs and a symmetric quantizer is

$$I_{B,L} = I(X;Z) = H(Z) - H(Z|X) = -\sum_{j=0}^{L-1} p_j \log_2(p_j) + \sum_{j=0}^{L-1} P_{0j} \log_2(P_{0j}) \qquad (3.11)$$

where Z denotes the output of the equivalent channel, i.e., the quantized version of the output of the BIAWGN channel Y, $p_j = \Pr[y_k \in T_j]$, and in the last equality we used the fact that due to the quantizer symmetry $P_{0j} = P_{1(L-1-j)}$. Since $\Pr[x_k = i] = 0.5,\ i = 0, 1$, $p_j = 0.5(P_{0j} + P_{1j}) = 0.5(P_{0j} + P_{0(L-1-j)})$, and so

$$\begin{aligned} I_{B,L} &= \sum_{j=0}^{L-1} P_{0j} \log_2(P_{0j}) + 1 - 0.5 \sum_{j=0}^{L-1} (P_{0j} + P_{0(L-1-j)}) \log_2(P_{0j} + P_{0(L-1-j)}) \\ &= 1 - \sum_{j=0}^{L-1} P_{0j} \log_2 \frac{P_{0j} + P_{0(L-1-j)}}{P_{0j}} \end{aligned} \qquad (3.12)$$

which agrees with the result of [77].

Similarly to the cutoff rate, the only reference that will be used in the next sections to compare the mutual information achieved by different quantizer designs, will be the unquantized BIAWGN mutual information $I_{B,\infty}$ which is

$$I_{B,\infty} = I(X;Y) = h(Y) - h(Y|X) = h(Y) - \frac{1}{2} \log_2 \left(2\pi e \sigma^2\right) \qquad (3.13)$$

where $h(Y)$ can be expressed as in equation (3.8).

C. Uniform Scalar Quantization

Due to the symmetry of the quantizer, the problem of optimizing the MSE $D_{B,L}$, the cutoff-rate $R_{B,L}$, or the mutual information $I_{B,L}$, with respect to the quantization boundaries $a_1, a_2, \ldots, a_{L-1}$, is $(\lceil L/2 \rceil - 1)$-dimensional. For uniform quantization, this problem can be made one-dimensional since $a_j = (j - L/2)\tau$, $j = 1, 2, \dots, L-1$, $\tau \in \mathbb{R}$, $\tau > 0$. In this Section, we design several L-level uniform scalar quantizers with step τ that either minimize the MSE $D^u_{B,L}$, or maximize the cutoff-rate $R^u_{B,L}$, or maximize the mutual information $I^u_{B,L}$. The superscript u is used to denote uniform quantization.

1. MSE Minimization

Uniform quantization in the MMSE sense is defined as having all boundaries a_j, $j = 1, 2, \dots, L-1$, spaced by the quantizer step τ and all reconstruction values v_j, $j = 0, 1, \dots, L-1$, spaced by the same τ [82], i.e., for the inner intervals the reconstruction values v_j, $j = 1, 2, \dots, L-2$, are the midpoints of the intervals $T_j = (a_j, a_{j+1})$, $j = 1, 2, \dots, L-2$. So $a_j = \left(j - \frac{L}{2}\right)\tau$, $j = 1, 2, \dots, L-1$ and $v_j = \left(j - \frac{L-1}{2}\right)\tau$, $j = 0, 1, 2, \dots, L-1$.

One way to determine the quantization step τ minimizing the MSE $D^u_{B,L}$ is to use the Lloyd-Max algorithm [82]. An equivalent way that we are going to use here is to set the derivative of the MSE $D^u_{B,L}$ with respect to τ equal to zero and find the solution of the resulting equation. It is shown in Appendix A that the resulting equation is

$$\sum_{j=0}^{L-1} \left(j - \frac{L-1}{2}\right) \left[\left(j - \frac{L-1}{2}\right) \tau \, P_{0j} - c_{0j}\right] = 0 \qquad (3.14)$$

where

$$c_{0j} \;=\; \begin{cases} P_{00} - \frac{\sigma}{\sqrt{2\pi}}\,\mathrm{e}^{-\frac{\left[\left(1-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} & \text{for } j=0 \\[2ex] P_{0j} + \frac{\sigma}{\sqrt{2\pi}}\left(\mathrm{e}^{-\frac{\left[\left(j-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} - \mathrm{e}^{-\frac{\left[\left(j+1-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}}\right) & \\ & \text{for } j=1,2,\ldots,L-2 \\[2ex] P_{0(L-1)} + \frac{\sigma}{\sqrt{2\pi}}\,\mathrm{e}^{-\frac{\left[\left(L-1-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} & \text{for } j=L-1 \end{cases} \tag{3.15}$$

The advantage of expressing the results in the form of a nonlinear equation in τ over using the Lloyd-Max algorithm to optimize τ in this case is that τ can be numerically determined directly, e.g., using the function `fzero` in MATLAB. It allows a more general approach to quantizer optimization based on different criteria, as similar nonlinear equations for τ will be determined in the next subsections. This approach gave the same scalar quantizer step τ as the Lloyd-Max algorithm. These τ's have been plotted in Fig. 34 (dotted curves) as a function of $E_s/N_0 = \frac{1}{2\sigma^2}$. The top dotted curve corresponds to the τ of an $L = 3-$level quantizer, the second dotted curve from the top to the τ of an $L = 4-$level quantizer and so on, down to the bottom dotted curve, which corresponds to the τ of an $L = 8-$level quantizer

Since the reconstruction levels v_j do not play an important role, we could extend the uniform scalar quantizer definition to uniformly spaced boundaries $a_j = \left(j - \frac{L}{2}\right)\tau,\ j = 1, 2, \ldots, L-1$, but with representation levels v_j corresponding to the centroids of the Voronoi regions $T_j = (a_j, a_{j+1})$. In Appendix A it is shown that in this case the nonlinear equation in the uniform quantizer step τ becomes

$$\sum_{j=1}^{L-1}(v_{j-1}-v_j)\left(v_{j-1}+v_j-2\left(j-\frac{L}{2}\right)\tau\right)\left(j-\frac{L}{2}\right)e^{-\frac{\tau j}{2\sigma^2}\left[(j-L)\tau-2\right]} = 0 \tag{3.16}$$

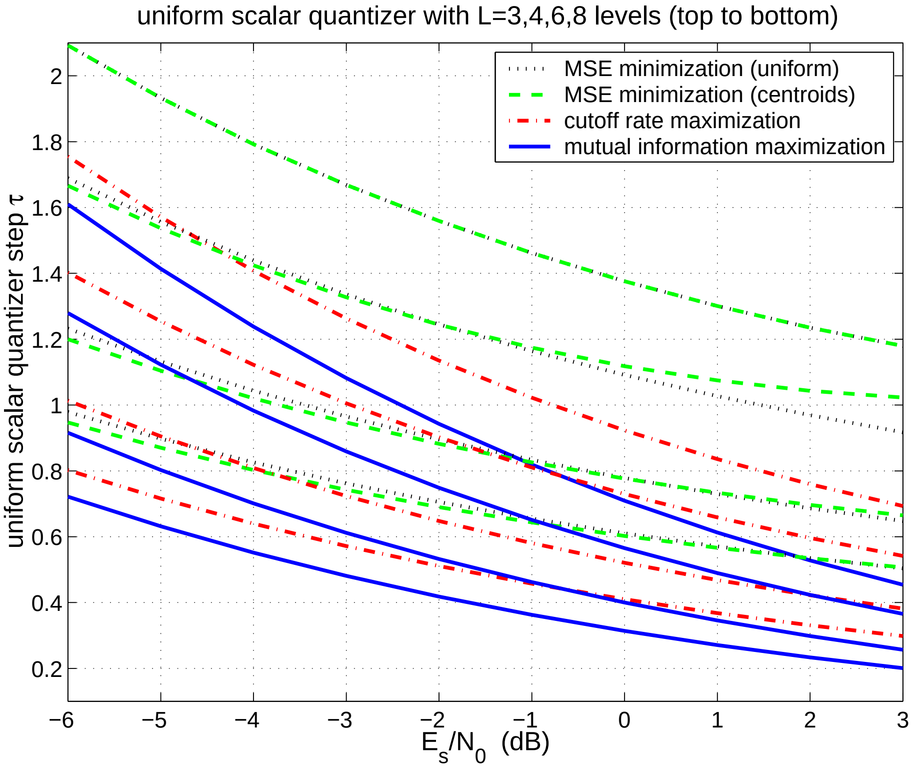

Fig. 34. The step τ of the uniform scalar quantizers that either minimize the MSE $D^u_{B,L}$, or maximize the cutoff rate $R^u_{B,L}$, or maximize the mutual information $I^u_{B,L}$, as a function of E_s/N_0 for $L = 3, 4, 6, 8$ levels.

where $v_j = \frac{c_{0j} - c_{0(L-1-j)}}{P_{0j} + P_{0(L-1-j)}}$, for $j = 0, 1, 2, \ldots, L-1$ and the c_{0j} are the same as in equation (3.15). These τ's have been plotted in Fig. 34 (dashed curves) as a function of $E_s/N_0 = \frac{1}{2\sigma^2}$. It can be seen in Fig. 34 that minimizing the MSE with either approach results in about the same uniform quantizer.

This is further confirmed in Fig. 35, where the MSE of both approaches has been plotted as a function of $E_s/N_0 = \frac{1}{2\sigma^2}$. For a fair comparison centroid quantization levels were used to determine the MSE of both quantizers. The MSE curves of the two quantizers almost coincide, i.e., the difference in the resulting MSE is not noticeable.

However, the MSE does not really have much to do with the limited precision

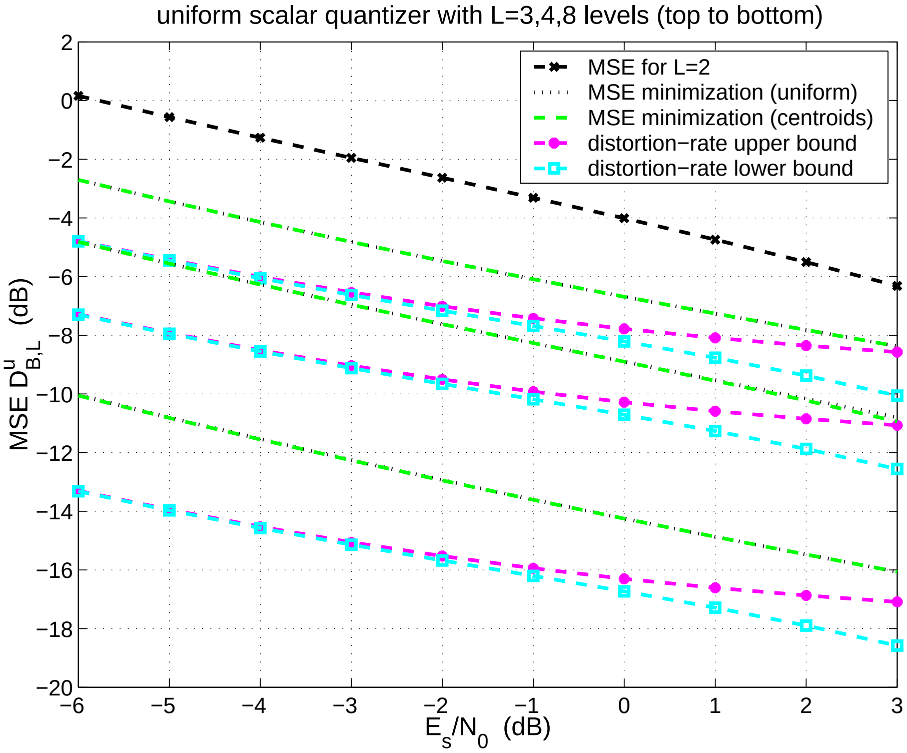

Fig. 35. The MSE $D^u_{B,L}$ of the uniform scalar quantizers that minimize the MSE $D^u_{B,L}$, as a function of E_s/N_0 for $L = 3, 4, 8$ centroid levels.

system performance, as we will see next. Actually, the only conclusion we can draw so far by comparing the uniform quantizer step τ of Fig. 34 for several quantization criteria, is that worse uniform MSE quantizers might be closer to the mutual information maximizing quantizers, i.e., they can result in higher information rates through the system than better uniform MSE quantizers.

Before proceeding with the other two criteria we give the scaled versions of Figs. 34 and 35 in Figs. 36 and 37, respectively. Figs. 36 and 37 show the scaled uniform quantizer step $\frac{\tau'}{\sqrt{N_0}}$ and the corresponding scaled MSE $\frac{D'^u_{B,L}}{N_0}$, respectively, in case the more practical model is used. In other words, Fig. 36 resulted from Fig. 34 by scaling with $\sqrt{\frac{E_s}{N_0}}$, since $\tau' = \tau\sqrt{E_s}$, and Fig. 37 resulted from Fig. 35 by scaling

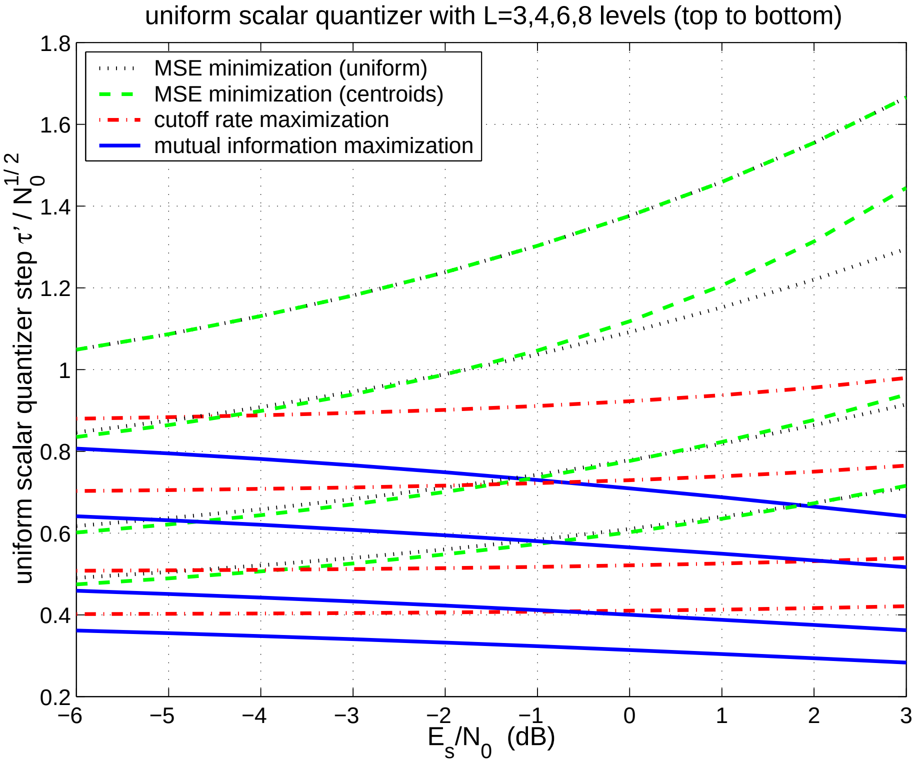

Fig. 36. The scaled step $\frac{\tau'}{\sqrt{N_0}}$ of the uniform scalar quantizers that either minimize the MSE $D^u_{B,L}$, or maximize the cutoff rate $R^u_{B,L}$, or maximize the mutual information $I^u_{B,L}$, as a function of E_s/N_0 for $L = 3, 4, 6, 8$ levels using the practical model.

with $\frac{E_s}{N_0}$, as $D'^{u}_{B,L} = E_s\, D^u_{B,L}$.

2. Cutoff Rate Maximization

Taking the derivative of the cutoff rate $R^u_{B,L}$ with respect to the uniform quantizer step τ and setting it equal to zero, we can get an equation for the step τ that maximizes the cutoff rate of the equivalent channel. The simplest form into which we could bring

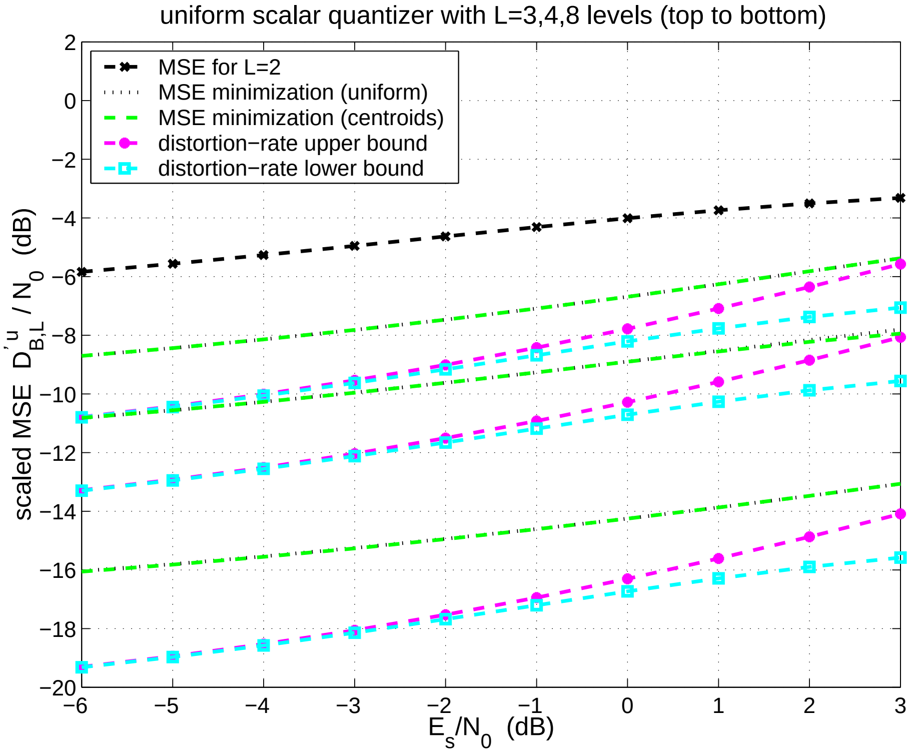

Fig. 37. The scaled MSE $\frac{D'^{u}_{B,L}}{N_0}$ of the uniform scalar quantizers that minimize the MSE $D^u_{B,L}$, as a function of E_s/N_0 for $L = 3, 4, 8$ centroid levels using the practical model.

the resulting equation is

$$\sum_{j=1}^{L-1} (\kappa_{j-1} - \kappa_j) \left(j - \frac{L}{2} \right) e^{-\frac{\tau j}{2\sigma^2} [(j-L)\,\tau - 2]} = 0 \qquad (3.17)$$

where $\kappa_j = \sqrt{\frac{P_{1j}}{P_{0j}}}$ for $j = 0, 1, \ldots, L-1$. The proof for this equation is given in Appendix B. Again, this last equation is nonlinear in τ, so its solution can only be determined numerically. Solving numerically this nonlinear equation for several L and SNRs, we always got a single numerical solution and this solution corresponds to the global maximum of the cutoff rate, although we have not been able to prove this observation.

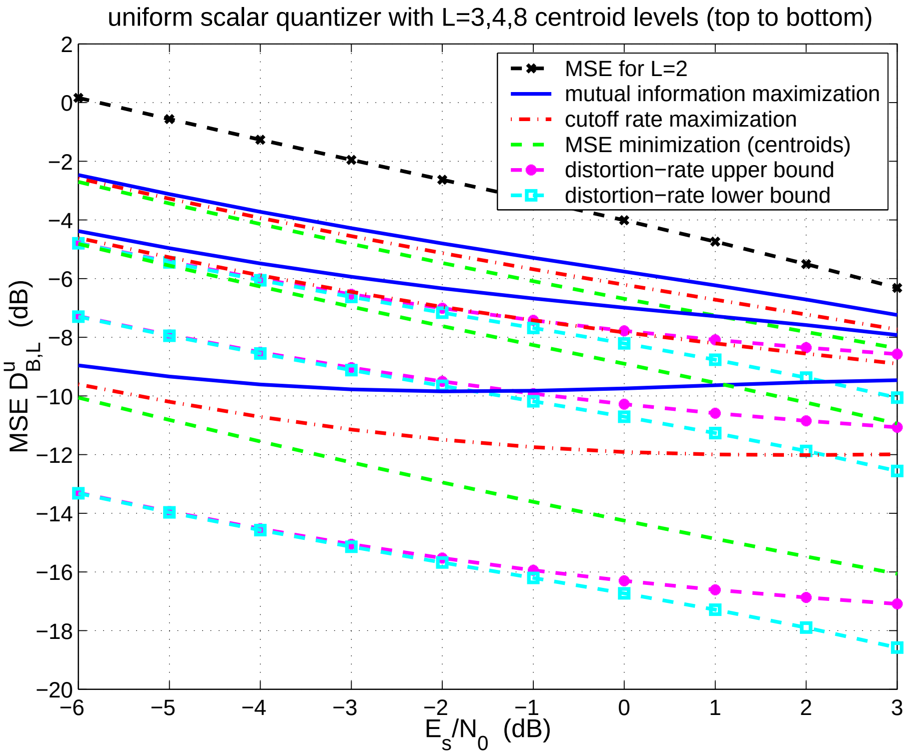

Fig. 38. MSE for quantized channel measurements based on MSE minimization, cutoff rate maximization and mutual information maximization as a function of E_s/N_0 for $L = 3, 4, 8$ centroid levels. The bounds on the minimum MSE of the BIAWGN channel output and the MSE of the $L = 2-$level BIAWGN quantized channel are also plotted.

The corresponding results for the cutoff rate maximizing quantization step τ are given in Figs. 34 and 36 with dash-dot curves. Our results exhibit a 3 dB difference in Fig. 34 from those in [63, 76, 79] due to the definition of $\frac{E_s}{N_0}$. Here, $\frac{E_s}{N_0} = \frac{1}{2\sigma^2}$, while in [63, 76, 79], $\frac{E_s}{N_0} = \frac{1}{\sigma^2}$. Another significant difference is that to our knowledge, this is the first time the practical model was combined with the cutoff rate maximization to observe that the corresponding uniform quantizer step remains almost constant, as Fig. 36 shows. Using the simplified model instead of the practical one in both [63, 79], this fact has never been noticed before.

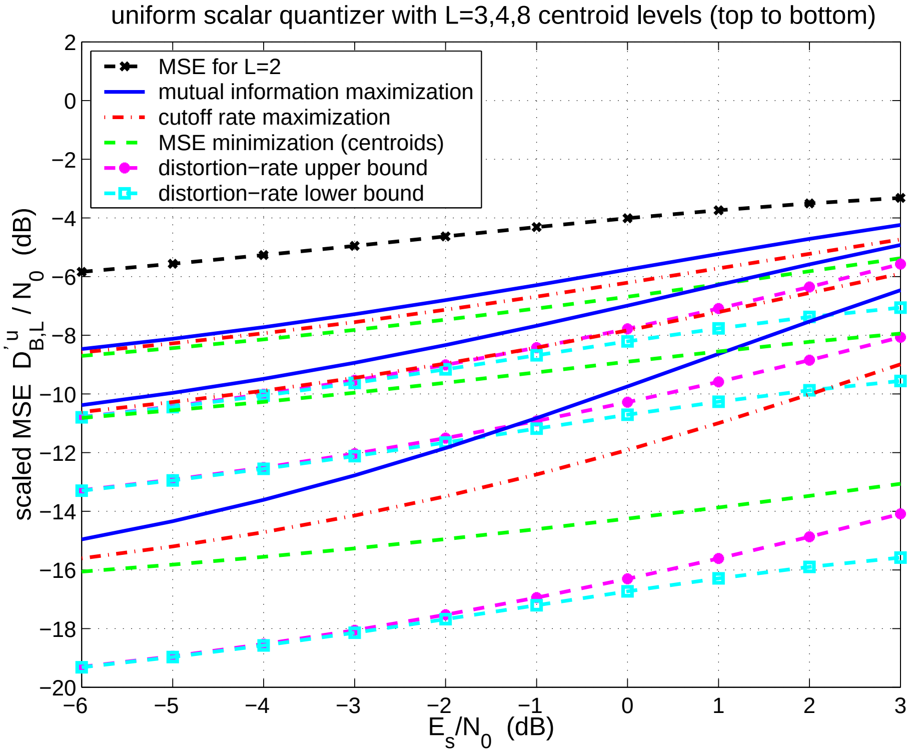

Fig. 39. Scaled MSE for quantized channel measurements based on MSE minimization, cutoff rate maximization and mutual information maximization as a function of E_s/N_0 for $L = 3, 4, 8$ centroid levels using the more practical model. The bounds on the minimum scaled MSE of the BIAWGN channel output and the scaled MSE of the $L = 2-$level BIAWGN quantized channel are also plotted.

The corresponding MSE, scaled MSE (for the practical model) and cutoff rate based on the quantizers of all the criteria considered have been plotted in Figs. 38, 39 and 40, respectively.

3. Mutual Information Maximization

Taking the derivative of the mutual information $I^u_{B,L}$ with respect to the uniform quantizer step τ and setting it equal to zero, we can get an equation for the step τ that maximizes the mutual information of the equivalent channel. The simplest form

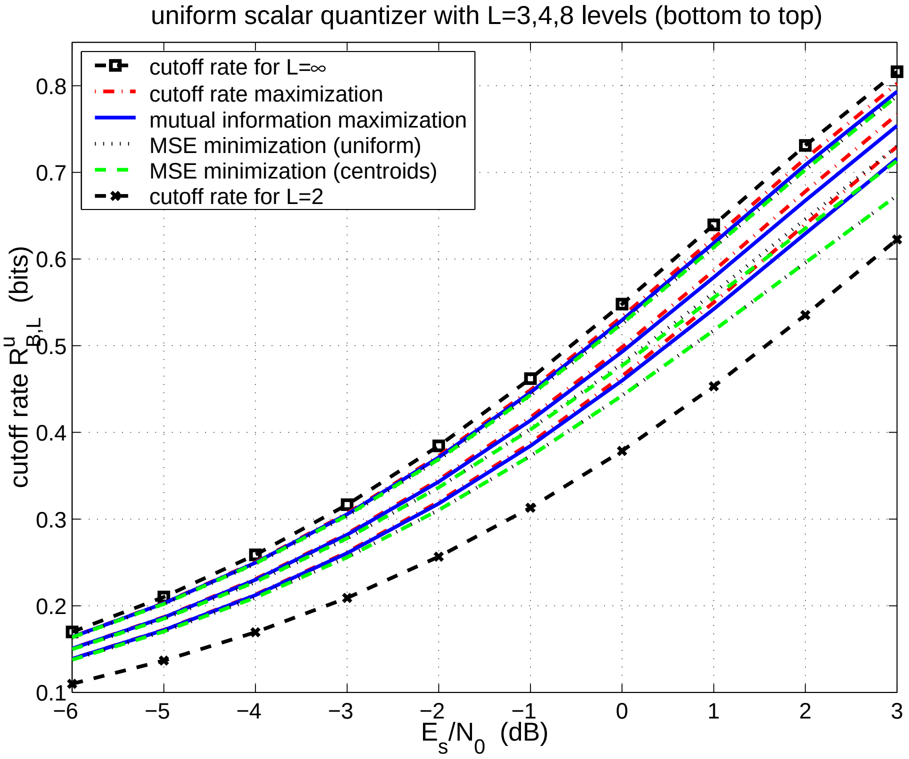

Fig. 40. Cutoff rate for quantized channel measurements based on MSE minimization, cutoff rate maximization and mutual information maximization as a function of E_s/N_0 for $L = 3, 4, 8$ levels. The cutoff rates of the unquantized BIAWGN channel ($L = \infty$) and of the $L = 2-$level BIAWGN quantized channel are also plotted.

into which we could bring the resulting equation is

$$\sum_{j=1}^{L-1} \left[\log\left(1+\kappa_{j-1}^2\right) - \log\left(1+\kappa_j^2\right)\right] \left(j - \frac{L}{2}\right) e^{-\frac{\tau j}{2\sigma^2}\,[(j-L)\,\tau-2]} = 0 \tag{3.18}$$

where $\kappa_j = \sqrt{\frac{P_{1j}}{P_{0j}}}$ for $j = 0, 1, \ldots, L-1$. The proof for this equation is very similar to the cutoff rate in Appendix B. Again, this last nonlinear equation in τ can only be solved numerically, and again, for several L and SNRs, we always got a single numerical solution that corresponds to the global maximum of the mutual information,

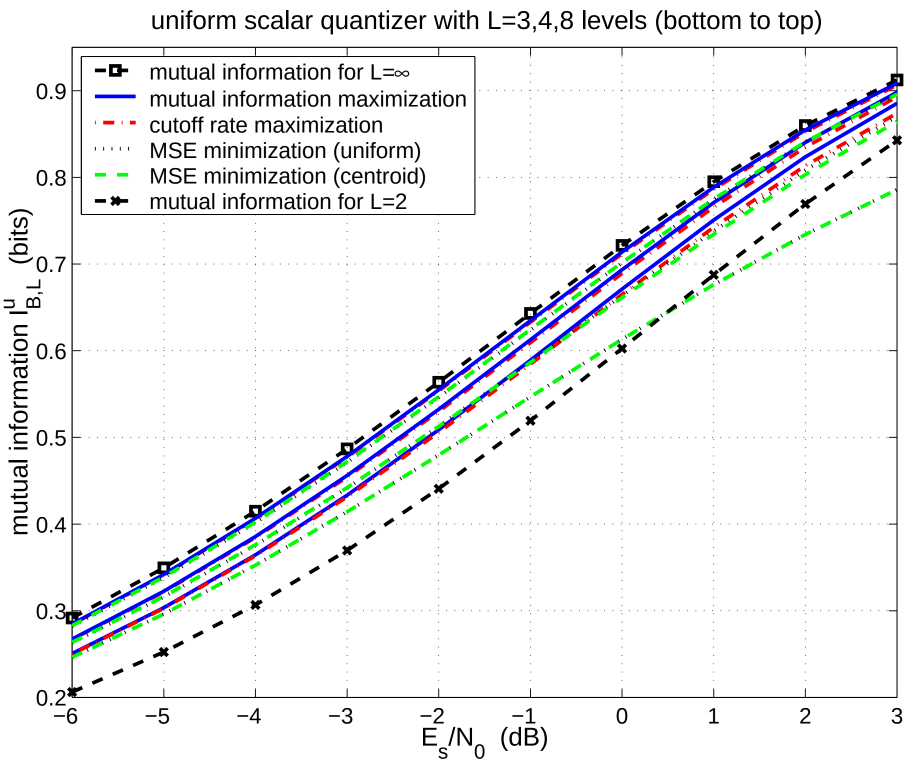

Fig. 41. Mutual information for quantized channel measurements based on MSE minimization, cutoff rate maximization and mutual information maximization as a function of E_s/N_0 for $L = 3, 4, 8$ levels. The mutual information of the unquantized BIAWGN channel ($L = \infty$) and of the $L = 2-$level BIAWGN quantized channel are also plotted.

although we have not been able to prove this observation.

The resulting τ's have been plotted in Figs. 34 and 36 with solid curves. The results in Fig. 34 agree with those in [78, 79], again with a 3 dB difference. Fig. 41 shows the mutual information between the input and the output of the equivalent channel with the different quantizers.

4. Comparison

From Figs. 34, 36, 40 and 41, we see that although the values of τ for cutoff rate maximization and mutual information maximization are different, the resulting difference of the corresponding mutual information and cutoff rate of the equivalent channels is negligible. This was also observed in [79] for $L = 3$. This means that both the cutoff rate and the mutual information can be used to design quantizers that allow about the same information rate through the limited precision system. This is not the case for the uniform quantizer designs based on the MSE. In fact, smaller MSE might actually incur a larger information rate loss.

Before proceeding with nonuniform scalar quantization and check whether similar conclusions hold in this case, we should make a remark about the robustness of uniform scalar quantization. So far we assumed that a different optimized quantizer is used for each $\frac{E_s}{N_0}$. But since for a fixed $\frac{E_s}{N_0}$ a quantizer with step τ anywhere between the mutual information maximizing τ and the cutoff rate maximizing τ is expected to yield about the same information rate and cutoff rate, the same uniform scalar quantizer is expected to exhibit good performance over a range of SNRs. The practical model is especially helpful when looking at robustness. The cutoff rate maximizing uniform quantizer is almost the same for all the SNRs we considered here (Fig. 36). In fact from Fig. 36 we can conclude that any constant quantizer step between the cutoff rate maximizing and the mutual information maximizing one is expected to do well over the whole range of SNRs considered in Fig. 36. The robustness of the channel measurements quantizer had only been noticed before in [72] by BER simulations of convolutional codes, but it is the first time the Fig. 36 has been used to justify such robustness through the cutoff rate, the mutual information and the practical statistical model. This quantizer robustness will be further confirmed later in this chapter through simulations.

D. Nonuniform Scalar Quantization

Due to the symmetry of the quantizer, the problem of optimizing the MSE $D_{B,L}$, the cutoff-rate $R_{B,L}$, or the mutual information $I_{B,L}$ with respect to the quantization boundaries $a_1, a_2, \ldots, a_{L-1}$, is $(\lceil L/2 \rceil - 1)$-dimensional for a nonuniform scalar quantizer. In the previous section we dealt only with uniform scalar quantization. This raises the question: how much better can we do with a nonuniform scalar quantizer? How much more is there to gain, if we allow for a more complex quantizer design? This section is motivated by these questions. We design several L-level nonuniform scalar quantizers that either minimize the MSE $D^{nu}_{B,L}$, or maximize the cutoff-rate $R^{nu}_{B,L}$, or maximize the mutual information $I^{nu}_{B,L}$. The superscript nu is used to denote nonuniform quantization.

1. MSE Minimization

This problem is the conventional nonuniform quantizer design problem where a distortion measure is involved, e.g., the mean squared error (MSE). In conventional nonuniform quantization both the boundaries and the quantization levels have to be optimized and this allows the use of the iterative Lloyd-Max algorithm for the nonuniform quantizer design, which is based on optimizing first the boundaries for given quantization levels and then the quantization levels given the boundaries and repeating until the process converges.

However, again to allow for a more general treatment that can be extended to the nonuniform quantizer design problems of the next two subsections, we will take all partial derivatives of the MSE $D^{nu}_{B,L}$ with respect to the $(\lceil L/2 \rceil - 1)$ different bound-

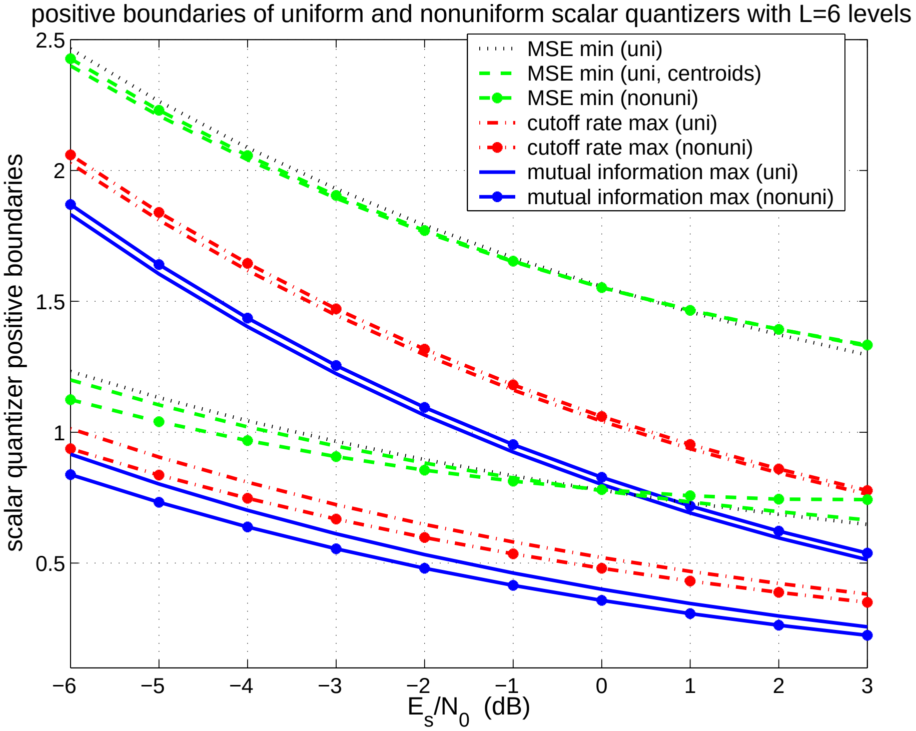

Fig. 42. The two positive boundaries a_4, a_5 of the uniform and nonuniform scalar quantizers that either minimize the MSE $D_{B,L}$, or maximize the cutoff rate $R_{B,L}$, or maximize the mutual information $I_{B,L}$, as a function of E_s/N_0 for $L = 6$ levels.

aries $a_1, a_2, \ldots, a_{\lceil L/2 \rceil - 1}$, assuming that the representation levels are the centroids v_j of the intervals $T_j = (a_j, a_{j+1})$. It can be shown following an approach similar to that of Appendix A that this approach yields the following system of nonlinear equations

$$v_{j-1} + v_j = 2\, a_j \ , \quad j = 1, 2, \ldots, \left\lceil \frac{L}{2} \right\rceil - 1 \tag{3.19}$$

where $v_j = \frac{c_{0j} - c_{0(L-1-j)}}{P_{0j} + P_{0(L-1-j)}}$, for $j = 0, 1, 2, \ldots, L-1$, and the c_{0j} are the same as in equation (3.6). These equations were expected since the nonuniform Lloyd-Max quantizer has centroid quantization levels and boundaries the midpoints between neighboring quantization levels.

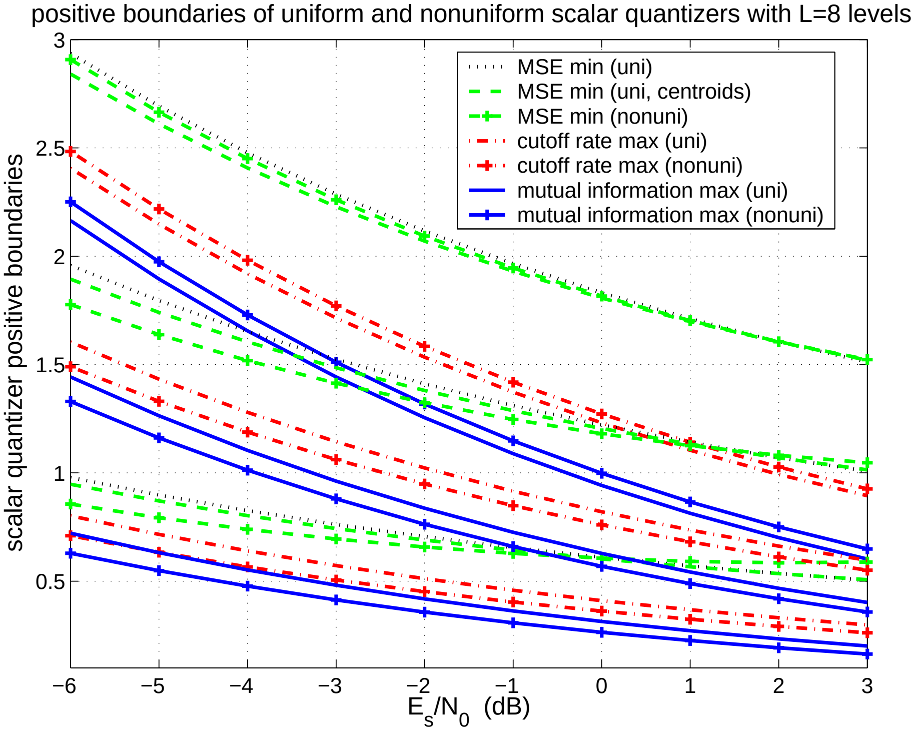

Fig. 43. The three positive boundaries a_5, a_6, a_7 of the uniform and nonuniform scalar quantizers that either minimize the MSE $D_{B,L}$, or maximize the cutoff rate $R_{B,L}$, or maximize the mutual information $I_{B,L}$, as a function of E_s/N_0 for $L = 8$ levels.

The solution of this nonlinear system of equations was determined using the MATLAB function `fsolve`. The positive boundaries a_j, i.e., $a_{\lfloor \frac{L}{2} \rfloor+1}, a_{\lfloor \frac{L}{2} \rfloor+2}, \ldots, a_{L-1}$, for $L = 6-$level and for $L = 8-$level quantization have been plotted in Figs. 42 and 43, respectively.

As can be seen from Figs. 42 and 43, as well as Fig. 44, which shows the MSE achieved by uniform and nonuniform scalar quantizers, nonuniform MSE minimizing quantization is almost uniform and achieves practically the same MSE as uniform MSE minimizing quantization.

Using the practical model to replot the results of Figs. 42, 43 and 44, we get

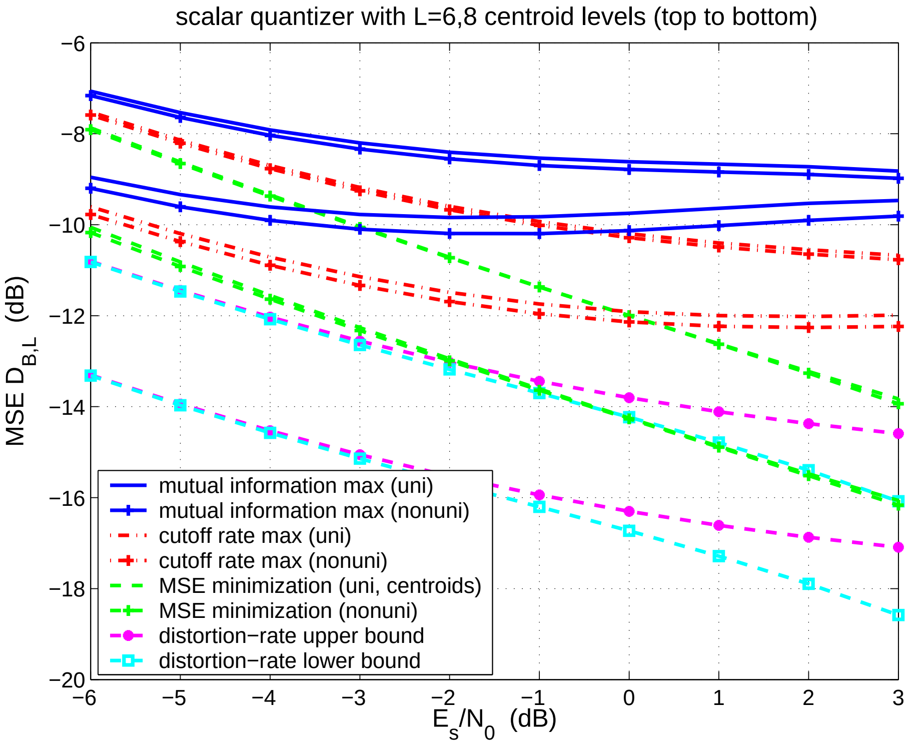

Fig. 44. MSE for uniform and nonuniform quantization of channel measurements based on MSE minimization, cutoff rate maximization and mutual information maximization as a function of E_s/N_0 for $L = 6, 8$ centroid levels. The bounds on the minimum MSE of the BIAWGN channel output are also plotted.

Figs. 45, 46 and 47, respectively.

2. Cutoff Rate Maximization

Here we are dealing with a nonconventional quantization problem where only the quantization boundaries have to be optimized and an iterative design procedure, such as the Lloyd-Max algorithm, cannot be applied. A design algorithm for nonuniform scalar quantizers based on the cutoff-rate maximization was introduced in [76]. This design algorithm can equivalently be expressed as a nonlinear system of equations, similarly to what was done for the MSE in the previous subsection. So, taking the

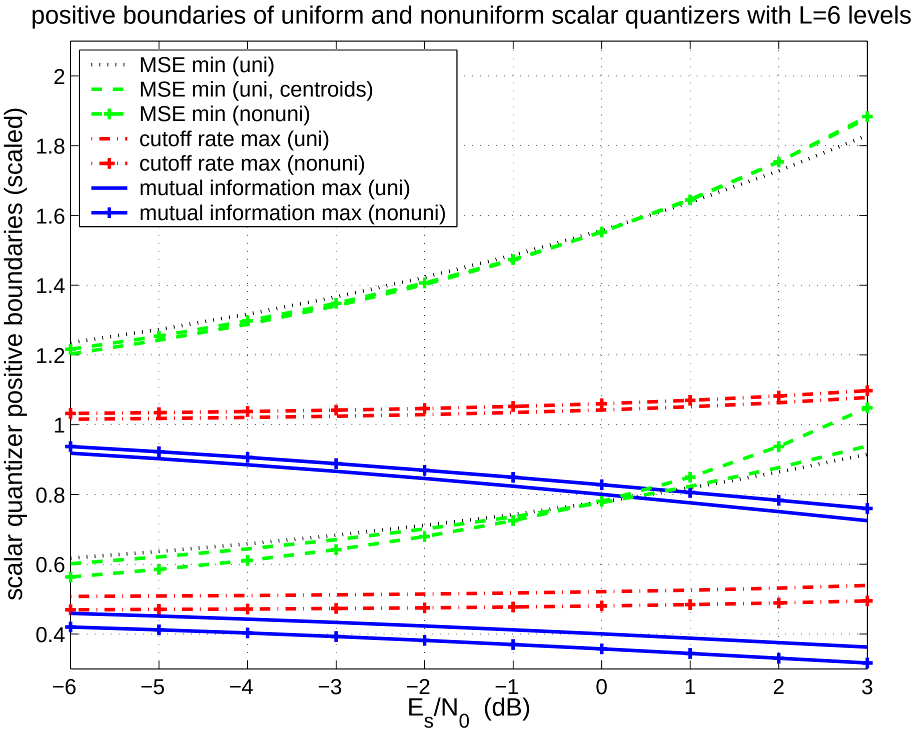

Fig. 45. The two positive boundaries (scaled) $\frac{a_4'}{\sqrt{N_0}} = a_4 \sqrt{\frac{E_s}{N_0}}$ and $\frac{a_5'}{\sqrt{N_0}} = a_5 \sqrt{\frac{E_s}{N_0}}$ of the uniform and nonuniform scalar quantizers that either minimize the MSE $D_{B,L}$, or maximize the cutoff rate $R_{B,L}$, or maximize the mutual information $I_{B,L}$, as a function of E_s/N_0 for $L = 6$ levels using the practical model.

partial derivatives of the cutoff rate $R^{nu}_{B,L}$ with respect to the $\lceil \frac{L}{2} \rceil - 1$ boundaries $a_1, a_2, \ldots, a_{\lceil \frac{L}{2} \rceil - 1}$ we get the following system of nonlinear equations [76]

$$\kappa_{j-1}\, \kappa_j = e^{-\frac{2\, a_j}{\sigma^2}} \ , \quad j = 1, 2, \ldots, \left\lceil \frac{L}{2} \right\rceil - 1 \tag{3.20}$$

where as in uniform quantization $\kappa_j = \sqrt{\frac{P_{1j}}{P_{0j}}}$.

The positive quantization boundaries $a_{\lfloor \frac{L}{2} \rfloor + 1}, a_{\lfloor \frac{L}{2} \rfloor + 2}, \ldots, a_{L-1}$ have been plotted in Figs. 42 and 43 for $L = 6-$level and for $L = 8-$level quantization, respectively. The $L = 8-$level nonuniform quantization results agree with those of [63, 76], again with the 3 dB difference. In Figs. 45 and 46 the positive quantization boundaries have

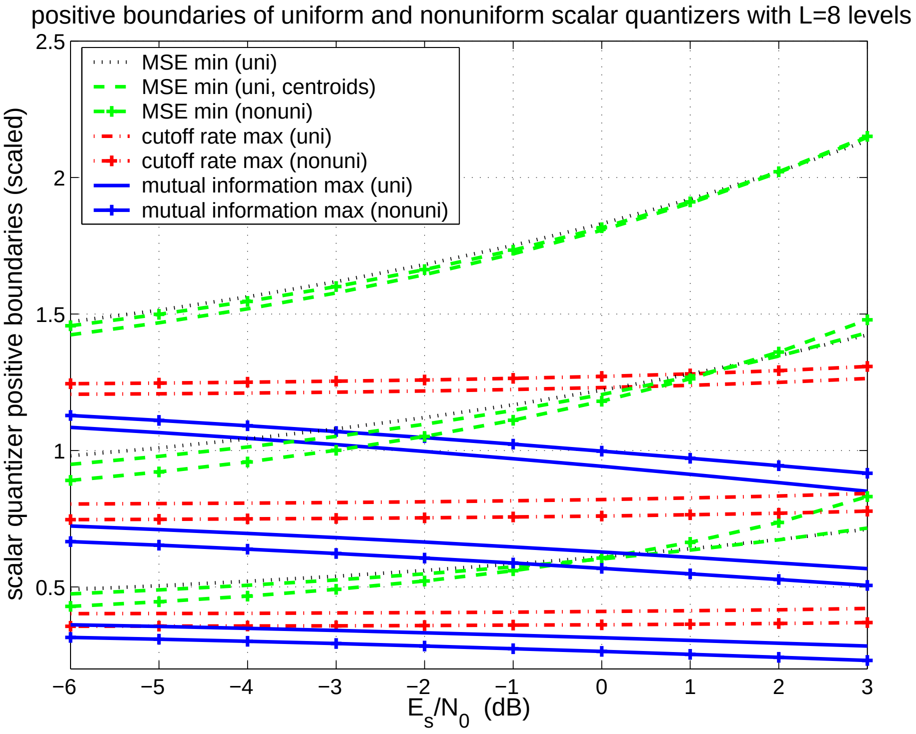

Fig. 46. The three positive boundaries (scaled) $\frac{a'_5}{\sqrt{N_0}} = a_5\sqrt{\frac{E_s}{N_0}}$, $\frac{a'_6}{\sqrt{N_0}} = a_6\sqrt{\frac{E_s}{N_0}}$ and $\frac{a'_7}{\sqrt{N_0}} = a_7\sqrt{\frac{E_s}{N_0}}$ of the uniform and nonuniform scalar quantizers that either minimize the MSE $D_{B,L}$, or maximize the cutoff rate $R_{B,L}$, or maximize the mutual information $I_{B,L}$, as a function of E_s/N_0 for $L = 8$ levels using the practical model.

been scaled and replotted using the practical model. The MSE and the scaled MSE of these quantizers are plotted in Figs. 44 and 47, respectively, and the cutoff rate of the equivalent channel corresponding to all the quantizers considered so far with $L = 6$ and $L = 8$ levels in Fig. 48. From all these figures one can see that the uniform and the nonuniform cutoff rate maximizing quantizers do not differ much from each other in terms of their boundaries (Figs. 42, 43, 45 and 46), the MSE between their input and output (Figs. 44 and 47) and the cutoff rate of the equivalent channel they define (Fig. 48). Fig. 49 shows that the corresponding mutual information of the input and

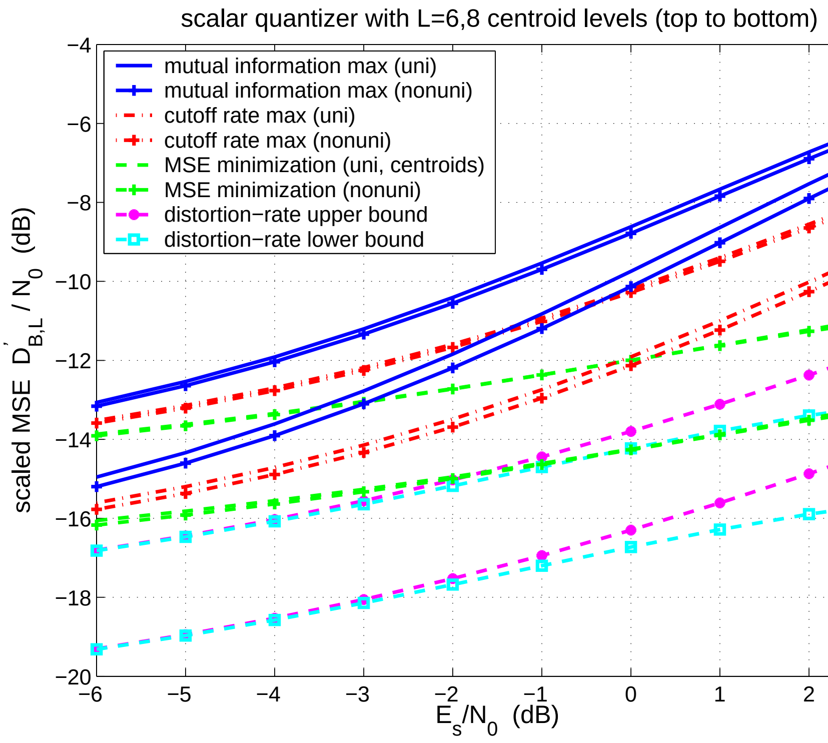

Fig. 47. Scaled MSE for uniform and nonuniform quantization of channel measurements based on MSE minimization, cutoff rate maximization and mutual information maximization as a function of E_s/N_0 for $L = 6, 8$ centroid levels. The bounds on the minimum MSE of the BIAWGN channel output are also plotted. For all curves the practical model has been used.

the output of the equivalent channel of both quantizers is also practically the same.

3. Mutual Information Maximization

Working similarly as in the previous subsection, i.e., taking the partial derivatives of the mutual information $I^{nu}_{B,L}$ with respect to the $\lceil \frac{L}{2} \rceil - 1$ boundaries $a_1, a_2, \ldots, a_{\lceil \frac{L}{2} \rceil - 1}$

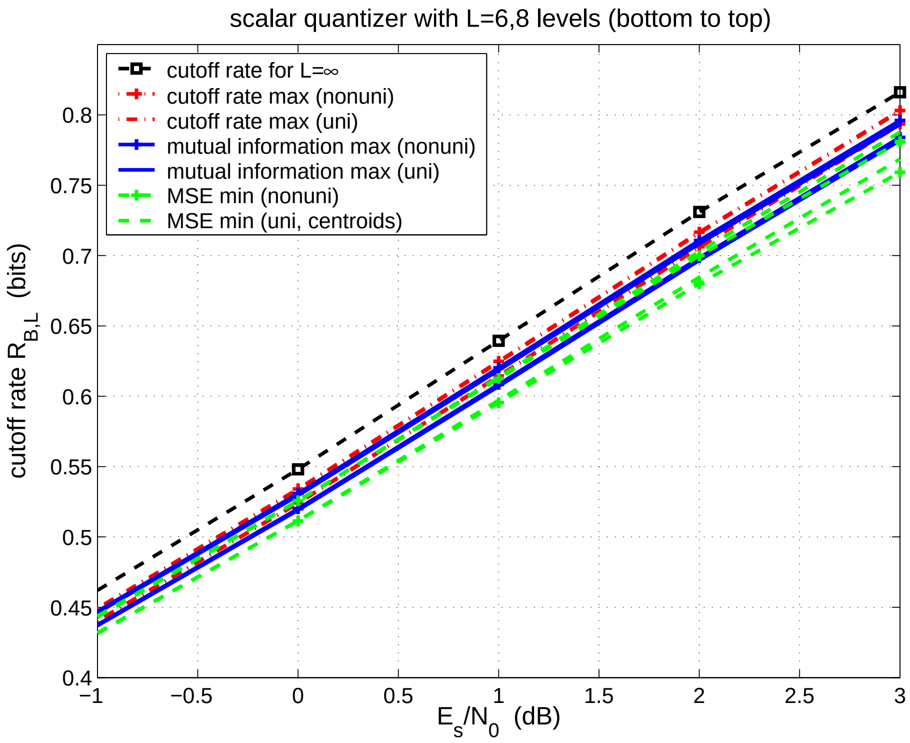

Fig. 48. Cutoff rate for uniform and nonuniform quantization of channel measurements based on MSE minimization, cutoff rate maximization and mutual information maximization as a function of E_s/N_0 for $L = 6, 8$ levels. The cutoff rates of the unquantized BIAWGN channel ($L = \infty$) are also plotted.

we can get the following system of nonlinear equations

$$e^{\frac{2a_j}{\sigma^2}} \left[\log\left(1+\kappa_{j-1}^2\right) - \log\left(1+\kappa_j^2\right)\right] = \log\left(1+\frac{1}{\kappa_j^2}\right) - \log\left(1+\frac{1}{\kappa_{j-1}^2}\right) \quad (3.21)$$
$$j = 1, 2, \ldots, \left\lceil\frac{L}{2}\right\rceil - 1$$

where as in uniform quantization $\kappa_j = \sqrt{\frac{P_{1j}}{P_{0j}}}$.

The positive quantization boundaries $a_{\lfloor\frac{L}{2}\rfloor+1}, a_{\lfloor\frac{L}{2}\rfloor+2}, \ldots, a_{L-1}$ have been plotted in Figs. 42 and 43 for $L = 6-$level and for $L = 8-$level quantization, respectively. In Figs. 45 and 46 the positive quantization boundaries have been scaled and replotted using the practical model. The MSE, the scaled MSE and the cutoff rate of these quantizer are plotted in Figs. 44, 47 and 48, respectively. The mutual information between the input and the output of the equivalent channel corresponding to all the quantizers considered so far with $L = 6$ levels is given in Fig. 49. From all these figures one can see that the uniform and the nonuniform mutual information maximizing quantizers do not differ significantly from each other in terms of their boundaries (Figs. 42, 43, 45 and 46), the MSE between their input and output (Figs. 44 and 47), the cutoff rate and the mutual information of the equivalent channel they define (Figs. 48 and 49).

4. Comparison

Comparing all the scalar quantizers we have considered so far, uniform or nonuniform, MSE minimizing, cutoff rate maximizing and mutual information maximizing, we can conclude that allowing a more complex design, i.e., nonuniform versus uniform, does not really yield noticeable gains neither in terms of the cutoff rate nor in terms of the mutual information. Another important fact is that improving the MSE through nonuniform MSE-minimizing quantization can result in lower mutual information and cutoff rate than uniform MSE minimizing quantization.

Regarding the robustness of the nonuniform quantizers, they exhibit similar robustness to their uniform counterparts. This can be seen in Figs. 45 and 46, as all the boundaries of a mutual information maximizing nonuniform scalar quantizer at a specific $\frac{E_s}{N_0}$ remain below the cutoff rate maximizing nonuniform quantizer boundaries. We expect that any scalar quantizer with boundaries between the respective cutoff rate maximizing boundaries and mutual information maximizing boundaries will not suffer any noticeable loss in terms of the mutual information or the cutoff rate of the equivalent channel. Furthermore, the nonuniform cutoff rate maximizing

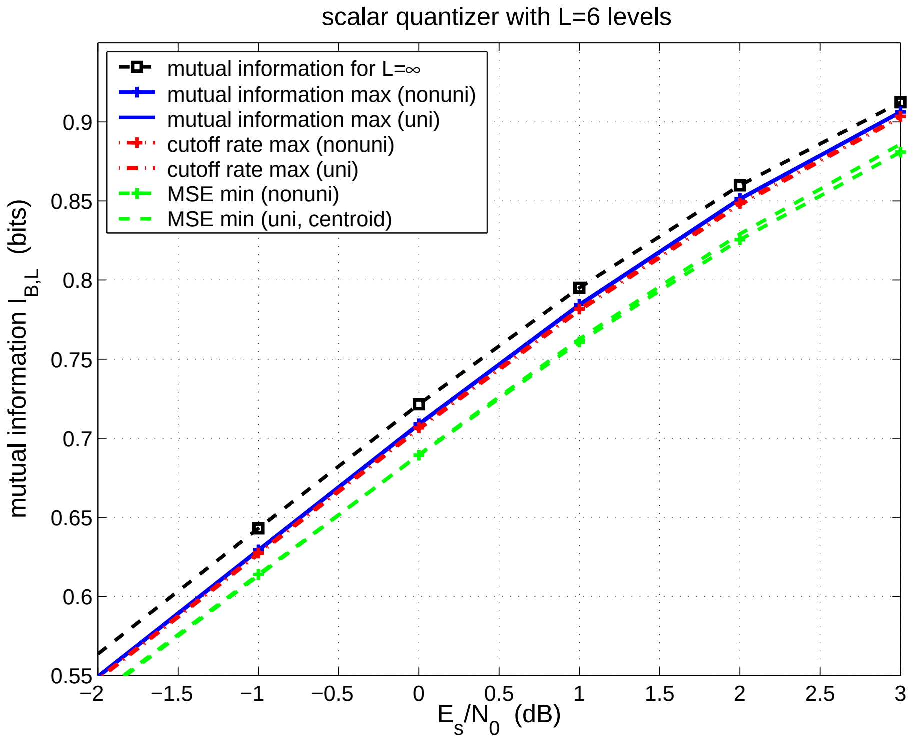

Fig. 49. Mutual information for uniform and nonuniform quantization of channel measurements based on MSE minimization, cutoff rate maximization and mutual information maximization as a function of E_s/N_0 for $L = 6, 8$ levels. The mutual information of the unquantized BIAWGN channel ($L = \infty$) is also plotted.

quantizers seem to be very robust, similarly to their uniform counterparts.

E. Simulation Results

In this subsection, we simulate low-density parity-check (LDPC) codes designed for the unquantized BIAWGN channel [20] with quantized channel measurements using MSE minimizing, cutoff rate maximizing and mutual information maximizing, uniform and nonuniform, quantizers and compare their performance. It is straightforward to see that the only difference in the message passing decoding algorithm when the output of the channel is quantized while the decoder can operate at infinite precision, lies in the channel log-likelihood ratios (LLRs). More precisely, when using L levels to quantize the channel measurements and the k^{th} unquantized received value y_k lies in the j^{th} quantization interval T_j, the corresponding channel LLR $L_{ch,k}$ is

$$\begin{aligned} L_{ch,k} &= \log \frac{\Pr\left[y_k \in T_j | x_k = 0\right]}{\Pr\left[y_k \in T_j | x_k = 1\right]} = \log \frac{P_{0j}}{P_{1j}} \\ &= \log \frac{P_{0j}}{P_{0(L-1-j)}}, \quad j = 0, 1, \ldots, L-1, \end{aligned} \tag{3.22}$$

where log is the natural logarithm. If $\delta(t)$ is the Dirac delta function, the probability density function of the channel LLRs conditioned on the transmission of the all-zeros codeword, is

$$P(l) = \sum_{j=0}^{L-1} P_{0j} \delta\left(l - f_j\right). \tag{3.23}$$

with $f_j = \log \frac{P_{0j}}{P_{0(L-1-j)}}$.

We consider two rate 1/2 LDPC codes, the regular (3,6) LDPC code and the simplest irregular code of [20]

$$\begin{aligned} \lambda(x) &= 0.234029x + 0.212425x^2 + 0.146898x^5 \\ &\quad +0.102840x^6 + 0.303808x^{19} \end{aligned} \tag{3.24}$$

$$\rho(x) = 0.71875x^7 + 0.28125x^8. \tag{3.25}$$

In Fig. 50, we can see how 2-bit and 3-bit uniform quantization with different criteria affect the decoding performance for the regular $(3,6)$ LDPC code. All simulation points assume either 100 frame errors or decoding of 2000 frames. From Fig. 50, first we can see that with 3-bit ($L = 8$) quantization the loss from the unquantized performance ($L = \infty$) can be as low as 0.1 dB. Furthermore, capacity maximization works slightly better than cutoff-rate maximization and both are significantly better

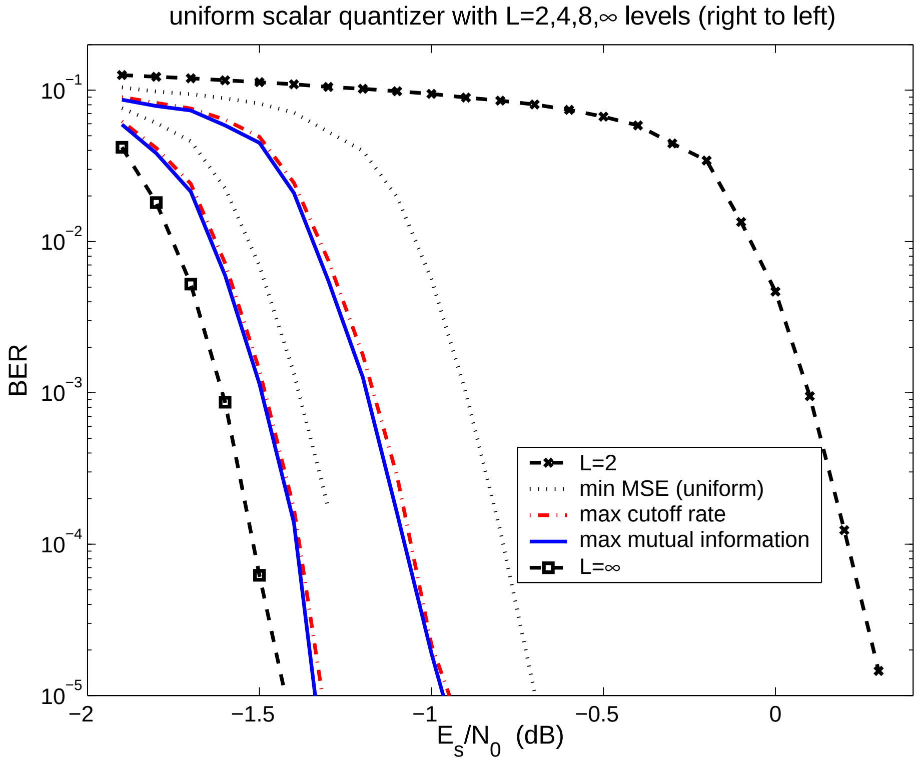

Fig. 50. Simulated BER for the length 10^4 regular (3,6) LDPC code with different uniform quantization criteria for 50 decoding iterations.

than MSE minimization.

A similar performance comparison is shown in Fig. 51 for the irregular code of (3.24) and (3.25). The code parameters are the same as those of Fig. 50 and from the simulation results, the superiority of mutual information maximization as a criterion for quantizer design is established.

In Figs. 52 and 53, the simulated performance of the regular and irregular LDPC codes with 3-bit quantization is plotted for all possible quantizers, i.e., both uniform and nonuniform for each quantization criterion and a robust uniform quantizer with step $\tau = 0.35\sqrt{N_0}$ for all simulated SNRs. This value for the robust quantizer was

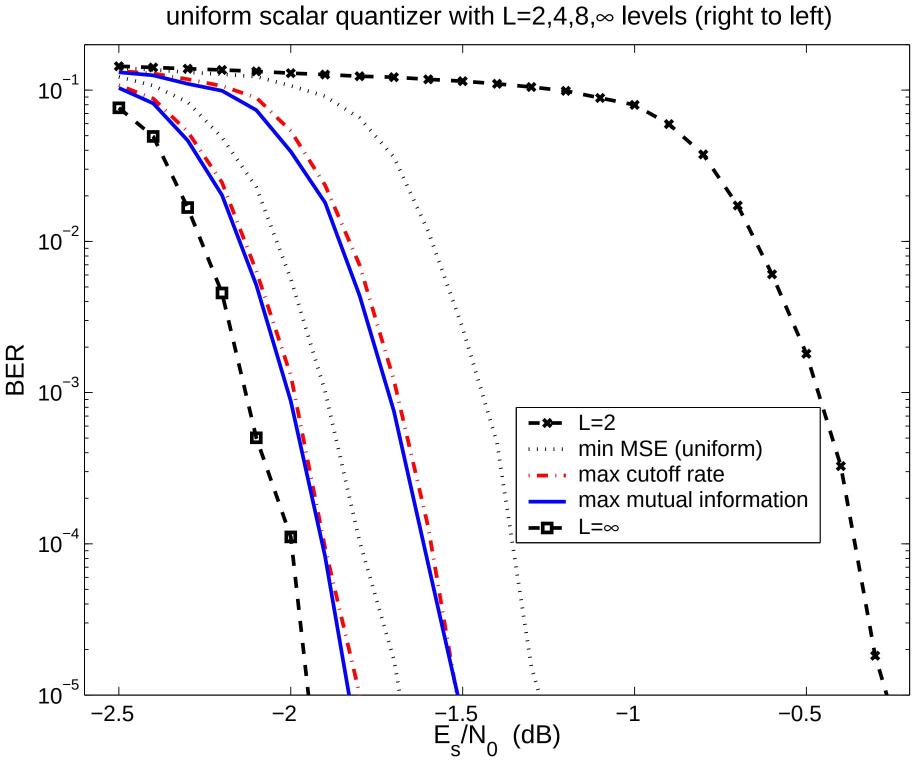

Fig. 51. Simulated BER for the length 10^4 irregular LDPC code of (3.24) and (3.25) with different uniform quantization criteria for 50 decoding iterations.

chosen as a value between the cutoff rate maximizing uniform quantizer step and the mutual information maximizing uniform quantizer step in Fig. 36 being a little closer to the latter as Figs. 50 and 51 show that there is a slight advantage in the performance of mutual information maximization.

Both Figs. 52 and 53, show that there is very little to gain by using nonuniform instead of uniform quantizers and there is almost no loss when the same uniform quantizer is used over a range of SNRs.

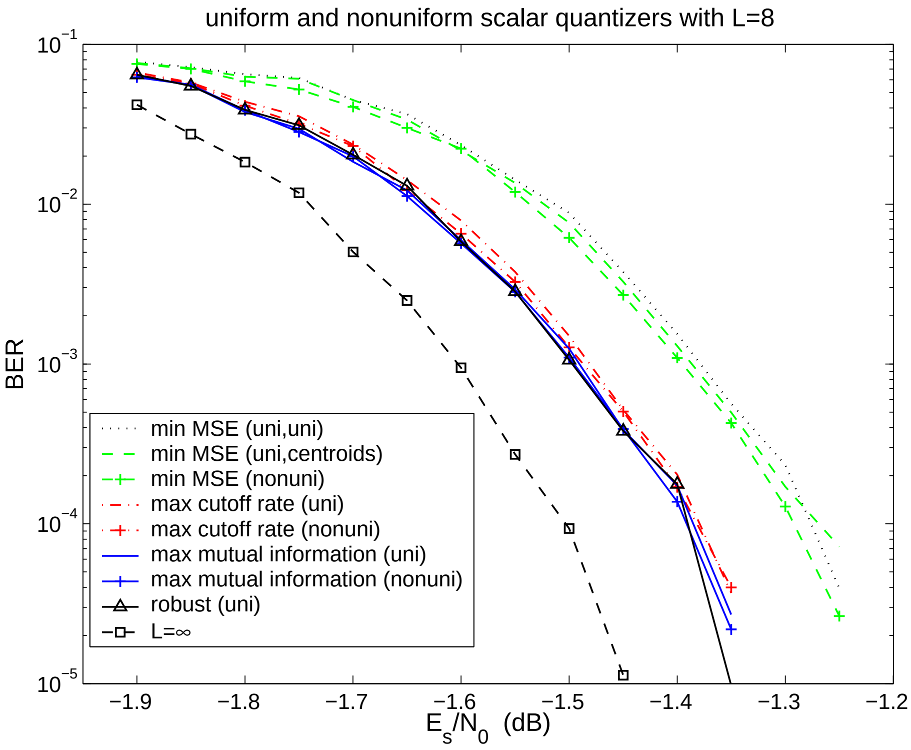

Fig. 52. Simulated BER for the length 10^4 regular (3,6) LDPC code with different uniform and nonuniform quantization criteria for 50 decoding iterations and $L = 8$ quantizer levels.

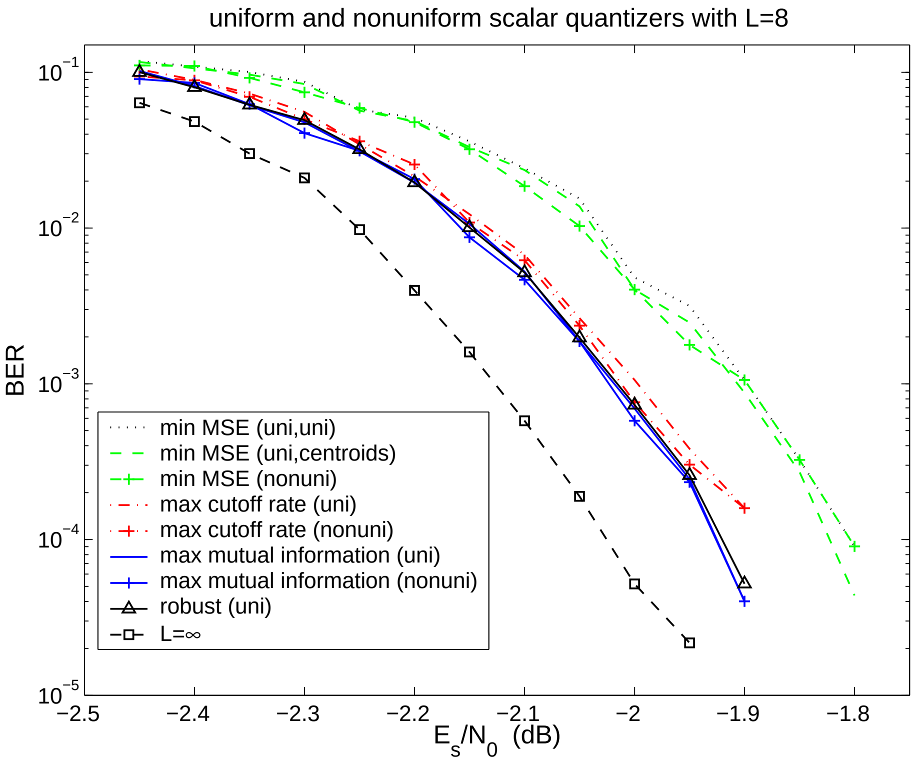

Fig. 53. Simulated BER for the length 10^4 irregular LDPC code of (3.24) and (3.25) with different uniform and nonuniform quantization criteria for 50 decoding iterations and $L = 8$ quantizer levels.

F. Conclusions and Discussion

In this chapter we studied several uniform and nonuniform scalar quantizers for quantization of coded channel measurements. The first conclusion of this study is that maximizing the mutual information or the cutoff rate of the equivalent channel yields better performance both in terms of the allowable throughput through the limited precision system and in terms of the BER of powerful channel codes, than minimizing the MSE between the input and the output of the quantizer. The difference between cutoff rate maximization and mutual information maximization is negligible.

The second conclusion is that the advantage of nonuniform scalar quantization over uniform scalar quantization is not noticeable, neither in the throughput nor in the BER. The last conclusion and probably the most important one for practical applications is that the same quantizer can be used over a wide range of SNRs without sacrificing in performance or throughput. This is observation is made here for the first time and is due to the more appropriate statistical model used.

Most of the above assumptions apply to almost all soft-input decoding channel codes, as the cutoff rate is directly related to the exponent of the union bound and so its maximization is expected to improve the channel decoding performance. The only channel codes that might be a little sensitive to the quantization effect are high rate codes. However, for most of the cases we think that our extensive comparison of all the criteria provided most of the answers to this quantization problem.

We believe there are only still two open directions remaining in this subject. The first is whether more complex quantization, i.e., variable-rate quantization [83] or nonscalar (vector) quantization [83], can yield significant gains. As we discuss in Appendix C, these gains, are mainly of theoretical interest, i.e., they will mainly serve as an answer to the still open problem of what the theoretical limits of the throughput through the limited precision system when complexity at the quantizer is not an issue.

The second open and more practical direction is how the extensive comparisons in this chapter can be extended to nonbinary modulation schemes and to different channels. The only results we are aware of for the case of any nonbinary modulation is in [84] and there has also been some work on quantization of PSK constellations [85, 86]. All this work assumes AWGN channels and cutoff rate maximization.

More work has been done for binary input Rayleigh fading channels [66, 67, 68, 69, 77, 78, 80, 81]. However, again an extensive study of all the quantization approaches is missing, as well as any type of quantizer design for nonbinary input Rayleigh fading channels.

CHAPTER IV

FASTER-THAN-NYQUIST SIGNALING[†]

Faster-than-Nyquist signaling (FNS) introduces intersymbol interference (ISI), but increases the bit-rate while preserving the signaling bandwidth. For sinc pulses, it has been established that with a small increase in the signaling-rate beyond the Nyquist rate there is no reduction in the minimum Euclidean distance for binary signaling. In this chapter we generalize these observations to the family of raised-cosine pulses. The structure of the error events that reduce the minimum distance is examined and constrained coding ideas are suggested that theoretically allow even faster signaling. Then we propose ways of practically achieving these gains by designing appropriate constrained codes and through equalization and iterative joint equalization and decoding (turbo equalization).

A. Introduction

We consider the communication system of Fig. 4. At the transmitter one information symbol a_k is generated every τ seconds, passed through a transmit filter $g(t)$ and sent through an additive white Gaussian noise (AWGN) channel. The matched-filter samples at the signaling rate, which form a sufficient statistic for the sequence estimation problem, are then

$$y(n\tau) = \sum_k a_k \, x\left((n-k)\tau\right) + n'(n\tau). \tag{4.1}$$

[†] ©2003 IEEE. Reprinted, with permission, from “Exploiting faster-than-Nyquist signaling”, Angelos D. Liveris and Costas N. Georghiades, *IEEE Trans. Commun.*, Sept. 2003, pp. 1502-1511.

where $x(t) = g(t) * g^*(-t)$ is assumed to be strictly bandlimited to W Hz, and $n'(t)$ is a zero-mean, Gaussian process with autocorrelation $R_{n'n'}(u) = x(u)N_0/2$.

For intersymbol interference (ISI) free transmission over an ideal AWGN channel of bandwidth W, Nyquist's criterion dictates that $x(i\tau) = 0$ for $i \neq 0$ and the signaling rate $1/\tau$ cannot exceed the *Nyquist rate* of $2W$ signals/s. Thus, to avoid ISI, $\tau \geq 1/(2W)$ with equality achieved only with ideal sinc pulses

$$x(t) = E_s \,\mathrm{sinc}\left(\frac{t}{T}\right) = E_s \,\frac{\sin(\pi t/T)}{\pi t/T}, \tag{4.2}$$

where $x(0) = E_s$ is the energy per transmitted symbol and since we assume baseband transmission throughout this paper, $T = 1/(2W)$. The sinc pulse is a special case of the general family of raised-cosine pulses

$$x(t) = E_s \,\mathrm{sinc}(t/T)\, \frac{\cos(\pi \alpha t/T)}{1-(2\alpha t/T)^2}, \tag{4.3}$$

with $0 \leq \alpha \leq 1$ being the roll-off factor and $W = (1+\alpha)/(2T)$ the bandwidth they occupy. ISI free transmission for these pulses means $\tau = T = (1+\alpha)/(2W) \geq 1/(2W)$.

In *faster-than-Nyquist signaling (FNS)* information symbols are transmitted at a rate higher than that suggested by the Nyquist criterion, i.e., $1/\tau > 2W$ and, therefore, ISI is unavoidable. Signaling above the Nyquist rate comes at the expense of higher receiver complexity and higher transmitted power since more information symbols are sent per second. Since it was first studied in [87], the usefulness of FNS has not yet been determined. Several receivers have been suggested, some implying the feasibility of FNS [88, 89] and other not considering it worthwhile [90, 91].

The asymptotically optimum performance with respect to sequence error probability is determined by the minimum Euclidean distance between all possible waveforms at the (noiseless) output of the channel [4, 92, 93, 94]. In the literature, minimum distance numerical results [94] and more exact analytical bounding methods

[95, 96] have already been used for sinc pulses to establish that signaling approximately 25% above the Nyquist rate does not reduce the minimum Euclidean distance for the binary case.

In this chapter we take another look at FNS by applying some of the recent powerful iterative equalization, turbo coding and constrained coding techniques to the problem [97, 98]. Our aim is to exploit FNS by converting some or all of the throughput gains into coding gains by use of coding with appropriate decoding/equalization techniques. In Section B the minimum distance results for sinc pulses are extended numerically to the more general raised-cosine pulses. In Section C, the error patterns that cause the minimum distance to decrease are examined in more detail and ways of avoiding their occurrence through constrained coding are suggested, thus extending the range of the minimum distance "gains". A simple code is then constructed based on a finite state machine (FSM) implementing the necessary constraint using results from [98, 99]. In Sections D and E the minimum distance gains achieved through FNS are practically translated into higher signaling rates by use of suboptimum equalization, or into coding gains by employing high rate single parity-check codes and turbo equalization. The performance in the first case comes very close to that of optimum antipodal ISI-free signaling over an AWGN channel and in the second case, to that of optimum decoding over an ISI-free AWGN channel. In Section F, the relationship between bandwidth and throughput of FNS is explored and in Section G it is shown that the sampling rate at the output of the matched-filter can be lower than the signaling rate of FNS. The conclusions sum up the chapter.

B. Minimum Distance Estimation

In the system of Fig. 4, let the row vector $\boldsymbol{a} = [a_k]$ represent the input sequence to the transmit filter and

$$s_{\mathbf{a}}(t) = \sum_k a_k g(t - k\tau) \tag{4.4}$$

the corresponding transmitted signal. It is easily established that the following Euclidean distance relates closely to the probability of mistaking a transmitted sequence $\boldsymbol{a}$ to some other sequence $\boldsymbol{b}$:

$$d^2(\boldsymbol{a}, \boldsymbol{b}) = \int_{-\infty}^{\infty} \left[s_{\boldsymbol{a}}(t) - s_{\boldsymbol{b}}(t)\right]^2 dt = \sum_k \sum_l e_k e_l x\left((k-l)\tau\right) = \boldsymbol{e}\boldsymbol{X}\boldsymbol{e}^t = d^2(\boldsymbol{e}) \tag{4.5}$$

where $e_k = a_k - b_k$ (e.g., $e_k \in \{0, \pm 2\}$ for the binary case), i.e., $\boldsymbol{e} = \boldsymbol{a} - \boldsymbol{b}$ is the input difference (error) sequence, and $\boldsymbol{X} = [x_{ij} = x((i-j)\tau)]$ is a Toeplitz matrix formed by the values of the overall pulse $x(t)$ at the sampling instants $n\tau$, $n \in \mathbb{Z}$. Using this last expression we can numerically estimate the minimum Euclidean distance for various signaling alphabets, pulse shapes and signaling rates over all difference sequences $\boldsymbol{e} \neq \mathbf{0}$. We define the normalized minimum Euclidean distance d_{min} as

$$d_{min} = \min_{\boldsymbol{e} \neq \mathbf{0}} \frac{d(\boldsymbol{e})}{2\sqrt{E_b}} \tag{4.6}$$

which determines to a large extent the performance of the optimum receiver that tries to combat the introduced ISI, especially for high signal-to-noise ratios [4, 92, 93, 94].[1] E_b is the energy per information bit and equals E_s only for uncoded binary transmission.

In this chapter we focus on raised-cosine pulses which are the most often used in practice, but the results can be extended to other bandlimited pulses, includ-

[1] Note that this definition of d_{min} is different by a factor of $\sqrt{2}$ from its usual definition in the associated literature as $d_{min}^2 = d^2/(2E_b)$.

ing quadratic and trapezoidal roll-off, for all of which (theoretically) ISI has infinite length. In this case, the results of [4, 92] do not apply as they assume finite ISI, but d_{min} can still be used to determine a lower bound on sequence error probability [93, 94]. Furthermore, as already established in [94, 95, 96] and from our numerical results (see the next section), the difference sequences minimizing the distance for small roll-off factor α have length of at most 12 for the region of interest, which is up to and a little beyond the point where the minimum distance starts decreasing.[2] The numerical results for uncoded binary signaling, are presented in Fig. 54 for three roll-off factors $\alpha = 0\%, 10\%, 20\%$. The minimum distance is plotted as a function of the bandwidth efficiency R/W, i.e., the ratio of the signaling rate $R = 1/\tau$ to the utilized bandwidth $W = (1+\alpha)/(2T)$ (measured in bits/sec/Hz). The ISI free signaling rate is $1/T$ corresponding to a bandwidth efficiency of $2/(1+\alpha)\,$bits/sec/Hz. The highest ISI-free bandwidth efficiency is $2\,$bits/sec/Hz, achieved by the ideal sinc pulse ($\alpha = 0$).

Fig. 54 suggests that an optimum detector does not suffer any distance loss up to signaling rates about 25% higher than Nyquist. For the sinc pulse, this was first noticed in [94] and then proven in [95, 96]. For nonzero roll-off factors the behavior of d_{min} is similar, but due to the excess bandwidth, the range of FNS rates without d_{min} reduction is more limited. That is why we only focus on the more bandwidth efficient raised-cosine pulses. Ways to extend this d_{min} behavior to higher rates by use of constrained coding are discussed in the following section. Then practical ways of exploiting the d_{min} gains of Fig. 54 are suggested in Sections D and E, to our knowledge for the first time.

[2] Actually, for all minimum distance plots all possible values of $\boldsymbol{e}$ of length 14 were considered for each point, i.e., $3^{14} \approx 4.8 \cdot 10^6$, which is enough to describe the d_{min} properties being exploited in our applications.

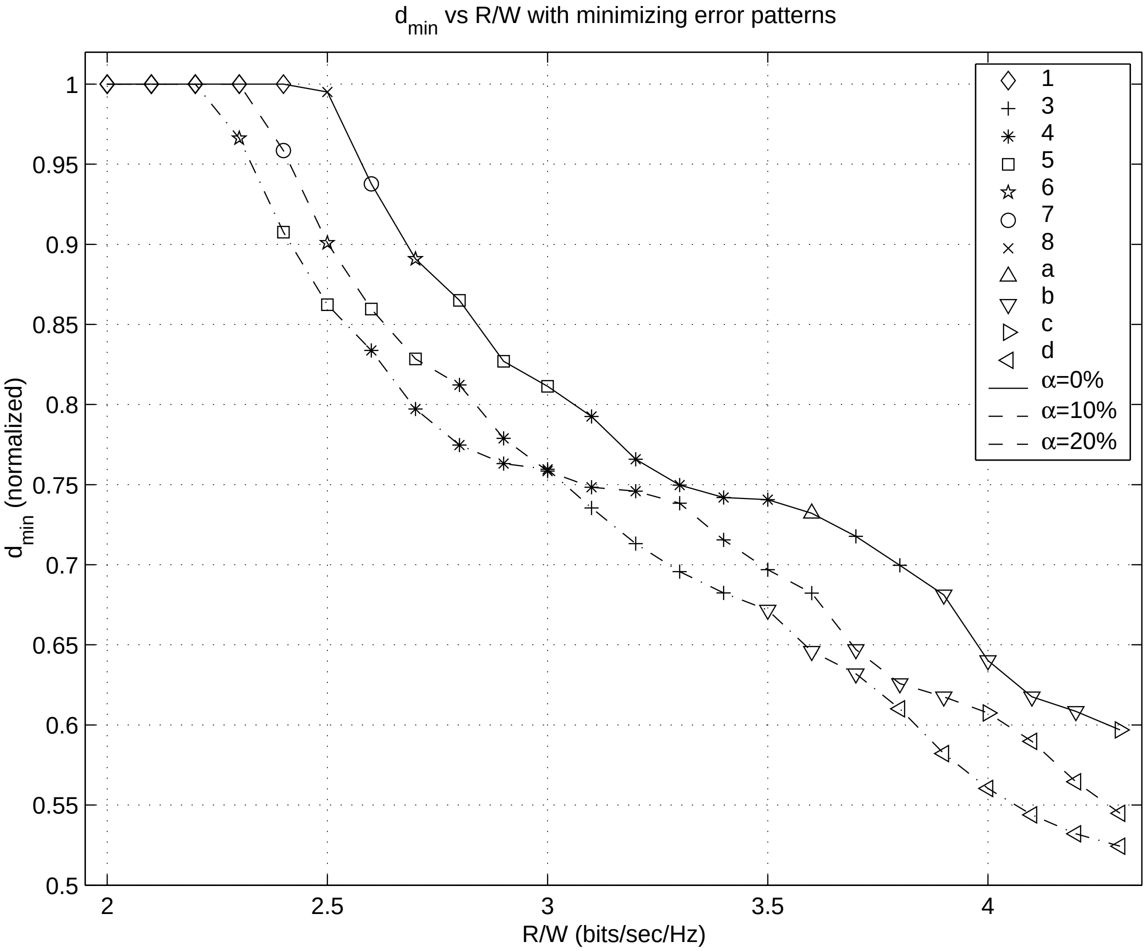

Fig. 54. Minimum Euclidean distance d_{min} as a function of the bandwidth efficiency for binary faster-than-Nyquist signaling (FNS) for the sinc pulse ($\alpha = 0\%$) and the raised-cosine pulses with roll-off $\alpha = 10\%, 20\%$. The error sequences $\boldsymbol{e}_{min}$ are also shown. The numbers show the length of the single alternating block $\boldsymbol{e}_{min}$ (e.g., $5 = \pm(\ldots 000 - + - + - 000 \ldots)$) and the letters stand for $\boldsymbol{e}_{min}$ containing more than one block ($a = \pm(\ldots 000 - + - 0 + - + 000 \ldots)$, $b = \pm(\ldots 000 {-}{+}0{-}{+}0{-}{+}000 \ldots)$, $c = \pm(\ldots 000 {-}{+}0{-}{+}00{-}{+}0{-}{+}000 \ldots)$, $d = \pm(\ldots 000 - +0 - +000 \ldots)$).

C. Error Events and Constrained Coding

In Fig. 54, as long as there is no change in d_{min} compared to the ISI-free case, the error sequence $\boldsymbol{e}_{min}$ that yields d_{min} is one with only a single nonzero component; i.e., a pair of input binary sequences is at d_{min} if the two sequences differ at a single position. Surprisingly, the error patterns $\boldsymbol{e}_{min}$ that first cause degradation in d_{min} in Fig. 54 contain a single long block of alternating polarity and, more generally, they consist of one or more blocks of alternating polarity, as Fig. 54 shows. This was also observed in [94, 95, 96] for sinc pulses. It seems that the FNS system is more vulnerable to higher frequency error events whose frequency content partly lies outside the utilized strictly bandlimited spectrum.

These error events $\boldsymbol{e}_{min}$ are depicted in Fig. 54 for all three roll-off factors. The numbers in the legend stand for error sequences containing a single block of alternating polarity of this length; for example, 5 represents the error pattern $\pm(\ldots 000 - + - + - 000 \ldots)$, where $+$ and $-$ correspond to $e_k = 2$ and $e_k = -2$, respectively. Each letter corresponds to an error pattern consisting of more than one block. Thus, $a = \pm(\ldots 000 - + - 0 + - + 000 \ldots)$, $b = \pm(\ldots 000 - +0 - +0 - +000 \ldots)$, $c = \pm(\ldots 000 - +0 - +00 - +0 - +000 \ldots)$, $d = \pm(\ldots 000 - +0 - +000 \ldots)$. Close to the "knee", the point where d_{min} starts decreasing, the d_{min} error event is a single block of alternating polarity, while for higher rates, first its length is reduced and then it starts splitting into two or three alternating blocks. Of course, it is obvious that such patterns can only arise from a pair of binary sequences that include alternating blocks of some length with opposite polarity.

Due to this structured form of error events and the associated binary sequences, constrained codes [98, 99] not allowing the occurrence of input sequences to the transmit filter that generate such error events, become very attractive. Employing such codes would keep d_{min} constant for even higher signaling rates, at the expense of some rate loss. Not allowing the occurrence of alternating blocks of length 3 and more in the error sequence, we repeat the numerical computations of d_{min} and e_{min} for this constrained case. The results are shown in Fig. 55.

The increase in d_{min} in Fig. 55 was combined with the code rate due to the constraints imposed on the input sequences to yield some gains over the unconstrained case. For the constraints imposed, one way of determining the maximum possible code rate used in Fig. 55 is by constructing a finite state machine (FSM) and determining its capacity. The FSM employed is depicted in Fig. 56(a) (higher rate FSMs that exclude longer sequences of alternating bits can similarly be obtained). The capacity of an FSM with r states is found from the *adjacency* matrix of the FSM, i.e., the $r \times r$ matrix $\boldsymbol{A}$ with elements a_{ij} equal to the number of edges from state i to state j, as the logarithm (base 2) of the largest real eigenvalue of $\boldsymbol{A}$ [98, 99]. For the FSM of Fig. 56(a), $\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ and thus, its capacity is 0.8114.

The constrained coding approach allows even faster signaling, around 50% FNS, at the expense of a small performance loss of around 1 dB compared to ISI-free signaling and slightly increased complexity.

However, the complexity of designing a code of high rate, close to capacity, is relatively high, especially when it is to be combined with a suboptimum detector trellis. Therefore, we start with a simple initial code design of rate 2/3 following the guidelines of [99], which has the form of Fig. 56(b), where some of the possible transitions between the FSM states have been dropped to yield a rate 2/3 code. The criteria for deciding which of the transitions to keep and which to drop, as well as for choosing the input-output mapping, still require further investigation and optimization.

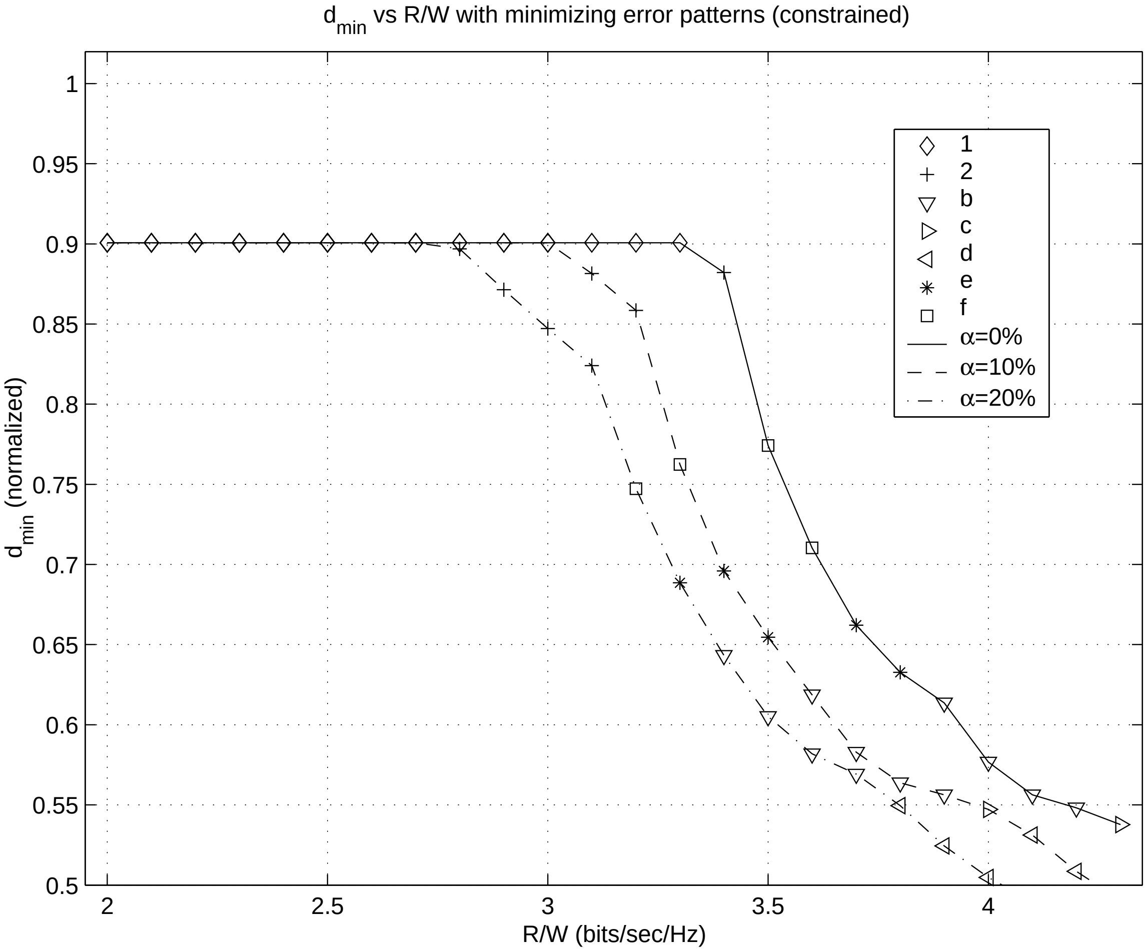

Fig. 55. Minimum Euclidean distance d_{min} as a function of the bandwidth efficiency for constrained binary FNS showing the error sequences $\boldsymbol{e}_{min}$. The numbers show the length of the single alternating block $\boldsymbol{e}_{min}$ (e.g., $2 = \pm(\ldots 000{-}{+}000\ldots)$) and the letters stand for $\boldsymbol{e}_{min}$ containing more than one block ($b = \pm(\ldots 000{-}{+}0{-}{+}0{-}{+}000\ldots)$, $c = \pm(\ldots 000{-}{+}0{-}{+}00{-}{+}0{-}{+}000\ldots)$, $d = \pm(\ldots 000 - +0 - +000\ldots)$, $e = \pm(\ldots 000 - +0 - +0 - +0 - +000\ldots)$, $f = \pm(\ldots 000 - +0 - +0 - +0 - +0 - +000\ldots)$).

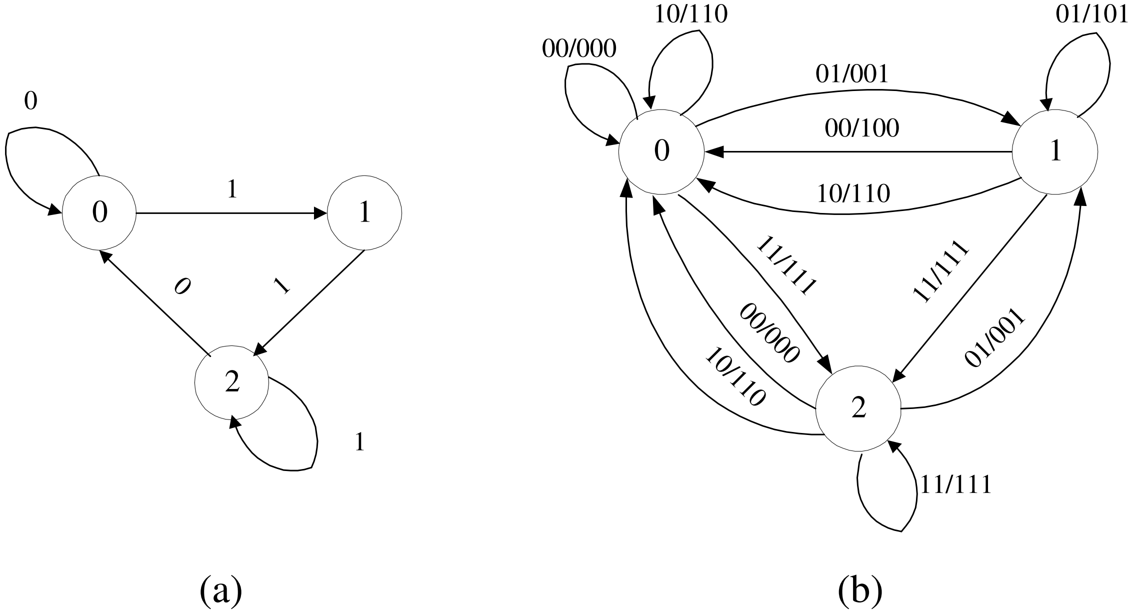

Fig. 56. (a) Finite state machine (FSM) not allowing binary input sequences which include the block 010. (b) Rate 2/3 encoder based on the FSM of Fig. 4(a).

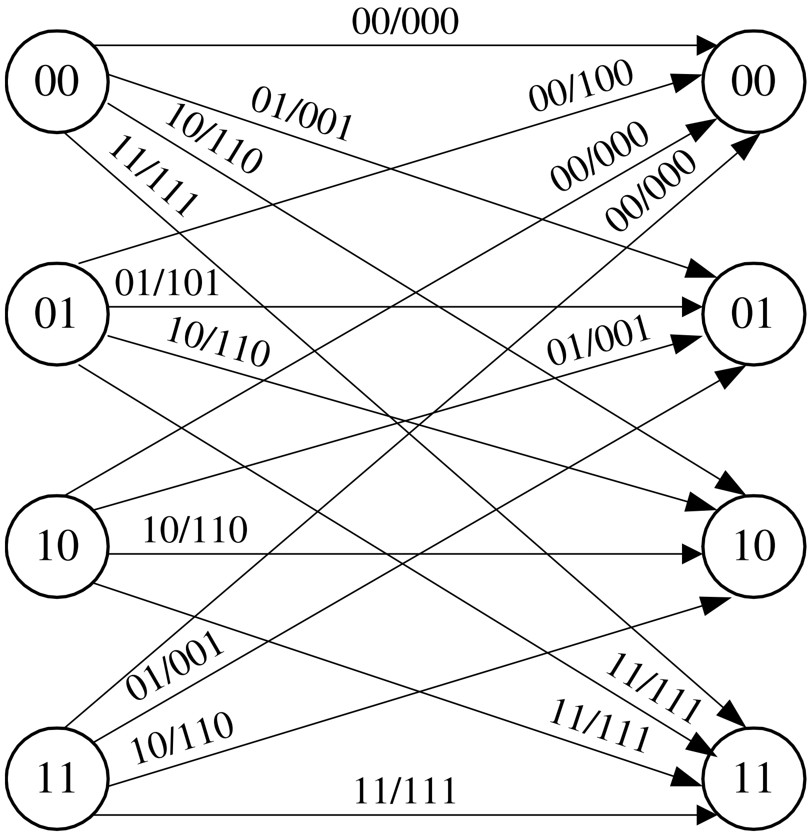

Fig. 57. Decoder trellis based on the FSM of Fig. 56(b).

Here we used systematic mapping, as this seems to yield a slightly smaller information bit error rate. Furthermore, the last two codeword bits over each transition, which constitute the systematic part, also determine the next state. So without loss of optimality, we can represent each state with the input leading to this state, i.e., state 1 with 01, state 2 with 11 and state 0 with either 00 or 10 or both. In the case that we use both 00 and 10 to represent state 0, the decoder trellis can take the form of Fig. 57.

We should note that it is possible to have a decoder trellis with only three states when using a single state to represent state 0, but the four-state trellis can be better combined with the detector trellis, discussed next, and this justifies the slightly increased complexity. In fact, it hardly increases the complexity, when combined with a reduced-state detection technique, as for small performance deviations from the optimum performance, the suboptimum scheme includes a reasonable number of states. Thus, the reduced-state detection trellis can contain the whole decoder trellis, as we will see next.

1. Detector Trellis and Simulation Results

We will use the trellis of Fig. 57 as the detector trellis in our scheme. We assume $\alpha = 10\%$, $\tau/T = 0.75$, i.e., $R/W = 2T/\tau/(1+\alpha) = 2.424\,\text{bits/sec/Hz}$, and the one-side duration of the raised-cosine pulse extends to $LT = 100T$, where T is the no-ISI signaling interval. For this case, from Figs. 54 and 55, we see that although the minimum distance starts decreasing, applying even the highest possible rate constrained code, i.e., a constrained code with rate equal to the capacity of the FSM of Fig. 56(a), does not yield any d_{min} gains. However, this refers to optimum detection and not to the suboptimum receiver structures actually employed. As the simulation results of Fig. 58 for systems with similar complexity show, even the rate 2/3 code can yield some gains, even after being penalized for its rate. The comparison is made with respect to the truncated version of Ungerboeck's Viterbi algorithm [4, 5] (see also Subsection 1 in Section D) and the improved truncated version employing decision feedback to cancel the residual ISI [6]. The performance is given in terms of information bit error rate as a function of E_b/N_0. All detectors of Fig. 58 operate directly on the matched-filter output samples and have about the same complexity in terms of number of transitions in a single trellis step. It is worth mentioning that incorporating decision feedback in the constrained coded system is expected to further improve its performance.

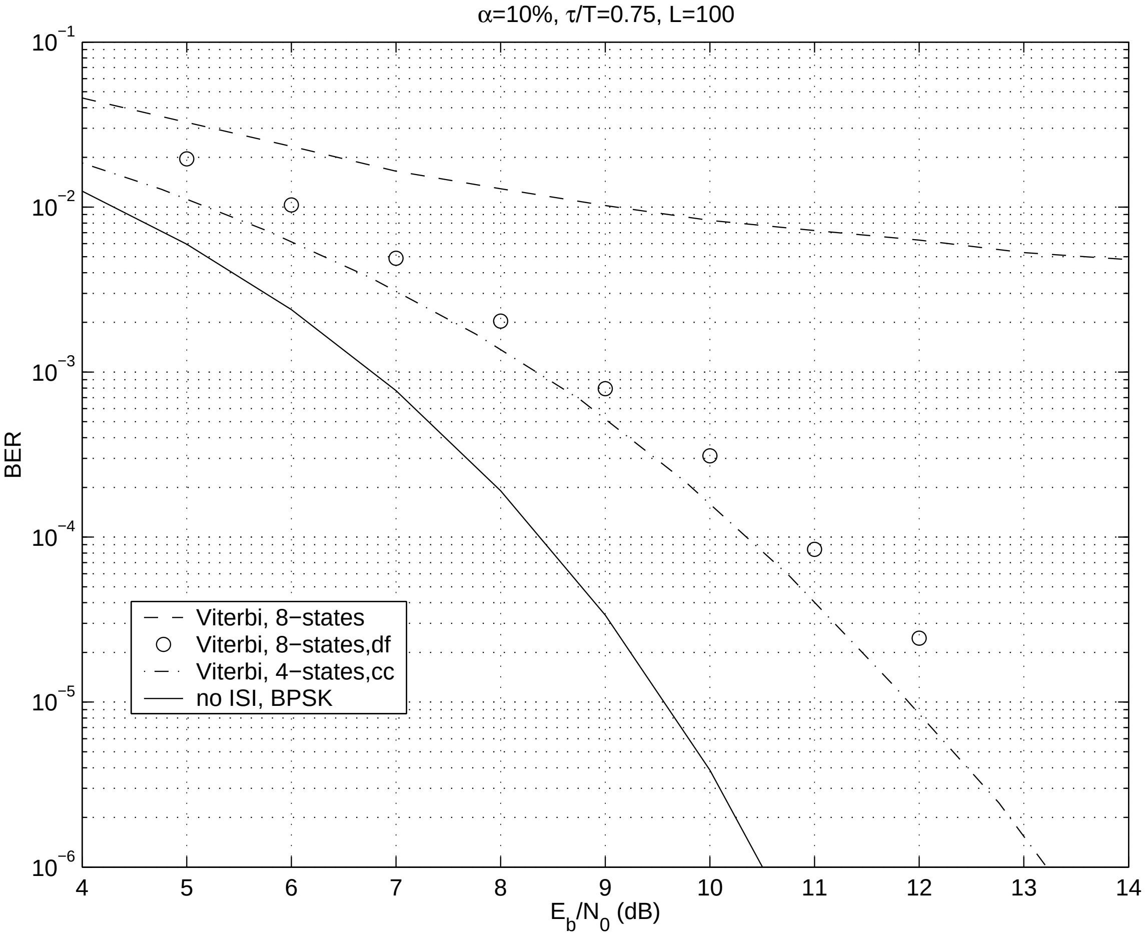

Fig. 58. Simulation results: Viterbi with 8 states is the truncated algorithm of [4, 5], Viterbi with 8 states and decision feedback (df) is truncated cancelling the residual ISI using the path history and tentative decisions [6] and the constrained coding (cc) scheme is based on the detector trellis of Fig. 57 (4 states but same number of transitions as the other two detectors) and has been penalized for its code rate.

Let us discuss in a little more detail the recursive processing in the reduced-state detector (plot marked “Viterbi, 4-states, cc” in Fig. 58). The trellis used is that of Fig. 57. This means that compared to the ordinary truncated detector based on [4], it is like performing three recursive computations for each transition, as each of them involves processing of three received symbols. Furthermore, there is some difference in the number of neighboring taps considered when canceling the ISI at each of the three symbols of each transition. This is because for the first of these three symbols the ISI of just the two past symbols indicated by the starting state of the transition can be cancelled, whereas for the second and third symbol the ISI of three and four, respectively, past symbols can be considered; apart from the two symbols of the starting state, the preceding symbols of the transition can also be considered. So the implemented recursive metric computation has the form

$$\begin{aligned}
J_{n+1}(a_{3n+1}, a_{3n+2}) &= J_n(a_{3n-2}, a_{3n-1}) + a_{3n}\left(y(3n\tau) - \sum_{k=1}^{2} x(k\tau)a_{3n-k}\right) \\
&\quad + a_{3n+1}\left(y\left((3n+1)\tau\right) - \sum_{k=1}^{3} x(k\tau)a_{3n+1-k}\right) \\
&\quad + a_{3n+2}\left(y\left((3n+2)\tau\right) - \sum_{k=1}^{4} x(k\tau)a_{3n+2-k}\right) \\
&= J_n(a_{3n-2}, a_{3n-1}) + \sum_{i=3n}^{3n+2} a_i y(i\tau) \\
&\quad - \sum_{i=3n}^{3n+2} a_i \sum_{k=1}^{i+2-3n} x(k\tau)a_{i-k}\,. \qquad (4.7)
\end{aligned}$$

It is straightforward to generalize this metric computation to a larger number of states and to higher rate codes.

Despite all the possible improvements that can be made to the constrained coded system, like incorporating decision feedback and using higher rate codes, for this low complexity it outperforms the other reduced-state detection schemes. The reason is that for such low complexity, not only the minimum distance error event but also other error events with a little larger distance affect the asymptotic performance. These can partly be predicted by looking at the error events that minimize the Euclidean distance for slightly faster signaling. For instance, we observed that the most frequent error events for large SNRs for the 8-state decision feedback scheme were those with a single block with length 5 and not those with length 6 or 7 as Fig. 54 indicates.

This changes when higher complexity can be tolerated, e.g., 64 states instead of 8. Then the optimum detector analysis is expected to be more precise and the constrained coding approach can only help as much as it was predicted. A practical system exploiting the gains for higher complexities or lower signaling rates would probably include some of the ideas of the systems proposed in the next sections that achieve the unconstrained gains.

D. Suboptimum Equalization

The focus of the previous section was how to prevent d_{min} from decreasing and how to use low complexity receiver structures that can work well at these increased FNS rates. The focus of this and the next section, however, has to do with exploiting the d_{min} gains that are shown at Fig. 54 at slightly lower FNS rates, i.e., at rates for which d_{min} is still the same as in the ISI-free case. The analysis was based on d_{min} between different transmitted sequences when employing the strictly bandlimited raised cosine pulse shape and from that we concluded for the binary case that there is no degradation in d_{min} at rates up to approximately 25% above the Nyquist when compared to the ISI-free case. However, although these gains have been discussed in the literature [94, 95, 96], no way of actually exploiting them has been suggested so far. This is the motivation for the work in this and the following section.

Referring to Fig. 4, the maximum-likelihood sequence detector selects the sequence $\boldsymbol{a}$ that maximizes the log-likelihood function

$$\ell(\boldsymbol{a}) = \sum_n a_n \, y(n\tau) - \frac{1}{2} \sum_k \sum_n a_n \, a_{n-k} \, x(k\tau). \tag{4.8}$$

If $\tau = T$, the system satisfies the Nyquist criterion, i.e., there is no ISI and the optimum detector is the sign-detector. However, when $\tau < T$, the Nyquist criterion is violated, ISI-free transmission is not possible and the optimum way of dealing with the ISI is maximum likelihood sequence estimation, which can be implemented with the Viterbi algorithm (VA) [4, 92] if the infinite ISI is truncated to some finite length.

Before proceeding with the equalizer structures, it is worth mentioning that a sampling rate at the output of the matched-filter lower than the signaling rate can be adequate. For strictly bandlimited pulses to W Hz, like the raised-cosine family, the matched filter output bandwidth is also strictly bandlimited to W Hz. Therefore, using interpolation, it is still optimum to process the Nyquist rate samples, i.e., sampling at $2W = (1+\alpha)/T$. The advantage would be a constant sampling rate even with varying FNS rate, which would be accounted for by adapting the coefficients in the optimum detector. The price paid would be a small loss due to the nonideal interpolation. The sampling rate is discussed in more detail in Section G.

We should also note that we already used some of the equalizer structures that will be presented next, in the previous section. But as Fig. 58 shows, these structures cannot outperform the coded performance and are quite far from the ISI-free performance for such high rates. Here the FNS rates are a little lower, thus the ISI-free performance can very closely be approached with these suboptimum schemes and this is the first reason they are studied in more detail in this section. The second reason is that these structures can be combined with decoding in an iterative receiver scheme allowing the translation of the increased signaling rates into coding gains.

1. The Truncated Optimum Equalizer

We use the modified VA (MVA) of [4] which allows operating directly on the matched-filter samples at the faster $1/\tau > 2W$ rate. If $y_n = y(n\tau)$ is the sample of the matched-filter output at time $t = n\tau$, L is the number of interfering symbols preceding the current symbol a_n and $J_n(a_{n-L+1}, \ldots, a_n)$ is the metric of the state $a_{n-L+1}, \ldots, a_n$ at time $t = n\tau$, the recursive relationship of the MVA is [4]

$$J_n(a_{n-L+1}, \ldots, a_n) = J_{n-1}(a_{n-L}, \ldots, a_{n-1}) + a_n \left(y_n - \sum_{i=1}^{L} x(i\tau)\, a_{n-i} \right), \quad (4.9)$$

where we dropped the common terms for all states. A trellis of 2^L states is needed for optimum equalization. The problem, of course, is that here the ISI is very long, theoretically infinite, so that the VA would be too complex. In the system of Fig. 4, we will use $\alpha = 10\%$[3] and truncate the theoretically infinite raised-cosine waveform $x(t)$ to $80T$ seconds ($40T$ before the peak and $40T$ after it), which is quite accurate for all practical purposes, i.e., the optimum VA should have more than $2^{40} \approx 10^{12}$ states.

The first step towards a suboptimum approach is to decrease L, the length of the ISI due to the preceding or following symbols considered by the VA, to a reasonable number [5] which will also depend on our system parameters α and $q \equiv \tau/T < 1$. The spectral efficiency of the FNS system is thus:

$$\frac{R}{W} = \frac{1/\tau}{(1+\alpha)/(2T)} = \frac{2}{q(1+\alpha)} \quad \text{bits/s/Hz.} \quad (4.10)$$

Thus, the percentage improvement, Δ, in spectral efficiency compared to a system

[3] All numerical results reported hereon are for $\alpha = 10\%$, chosen as a value used in high efficiency systems.

operating at no ISI is

$$\Delta = \frac{1-q}{q} \times 100\%. \tag{4.11}$$

For a roll-off factor $\alpha = 10\%$, which we will use throughout this and the next section, the spectral efficiency of the ISI-free system is 1.82 bits/s/Hz. We will consider two different τ's, $9T/10$ and $5T/6$, which correspond to spectral efficiencies $R/W = 2.02$ and $R/W = 2.18$ bits/s/Hz, i.e., 1% and 9% higher than the Nyquist rate and $\Delta = 11\%$ and $\Delta = 20\%$ higher than their ISI-free rate, respectively. For both these bandwidth efficiencies, there is no d_{min} reduction, as shown in Fig. 54. For $\alpha = 10\%$ and $\tau = 9T/10$ and $\tau = 5T/6$, the neighboring four symbols from each side contribute more to the ISI than the more distant ones. Thus, we form a truncated modified VA (TMVA) with 16 states ($L = 4$).

The general performance analysis of the TMVA can be found in [5], but the general expression is quite involved. A simpler way to approximate the performance of this detector is to assume that the residual ISI (RISI) [5]

$$\mathcal{I}_n = \sum_{|k|>L} a_{n-k} x(k\tau), \tag{4.12}$$

behaves almost like additive Gaussian noise, since it is a linear combination of a large number of independent zero-mean unit variance random variables a_k. The variance of this almost Gaussian random variable $\mathcal{I}_n$ is

$$\begin{aligned} \rho = E[\mathcal{I}_n^2] &= E\left[\sum_{|i|>L}\sum_{|k|>L} a_{n-i}a_{n-k}x(i\tau)x(k\tau)\right] &(4.13)\\ &= \sum_{|k|>L} x^2(k\tau) = 2\sum_{k>L} x^2(k\tau). &(4.14)\end{aligned}$$

Convergence in the sum above is guaranteed by the convergence of $\sum_{k>0}|x(k\tau)|$.

This "noise" term (the residual ISI) results in an increase in the overall noise variance and thus worsens the theoretically optimum performance $Q\left(\sqrt{2E_b/N_0}\right)$, where $E_b/N_0 = x(0)/N_0$ in our system. The modified argument in the Q-function using ρ will consider the increased noise variance $x(0)N_0/2+\rho$, i.e., $E_b/N_0' = x(0)/(N_0+2\rho/x(0))$, and so the approximate error probability for the truncated modified Viterbi detector will be

$$Pr(error) \approx Q\left(\sqrt{\frac{2E_b/N_0}{1+2(E_b/N_0)\rho/x^2(0)}}\right). \tag{4.15}$$

This last approximate expression not only shows that the truncated equalizer exhibits, as expected, worse performance compared to the optimum (no-ISI performance), but also that it might exhibit an error floor at low probabilities of error. Through ρ, the value of the SNR where the error floor effect starts degrading the performance significantly is a function of both L and τ. And although the situation becomes better for increasing L (lower error floor), the dependence on τ yields several local minima and maxima. The degradation in the performance can be seen in the simulation results of Figs. 59 and 60, when comparing the TMVA curve with the ISI free antipodal signaling curve (marked "uncoded, no ISI"). In these plots one can also see that the approximate expression (4.15) is quite close to the simulated performance.

2. Nearly Optimum Equalization

The performance loss of TMVA can be reduced by taking the residual ISI (RISI) into account. We name the filter that cancels the RISI the RISI canceller (RISIC). The idea and the name are borrowed from the interference cancellation techniques in multiuser detection systems. The first operation performed by the RISIC is to form a RISI estimate by passing the output of the TMVA through a linear filter whose tap weights are the same as the ISI tap weights, apart from the nine central ones that have already been accounted for in the TMVA (four taps from each side of the central tap

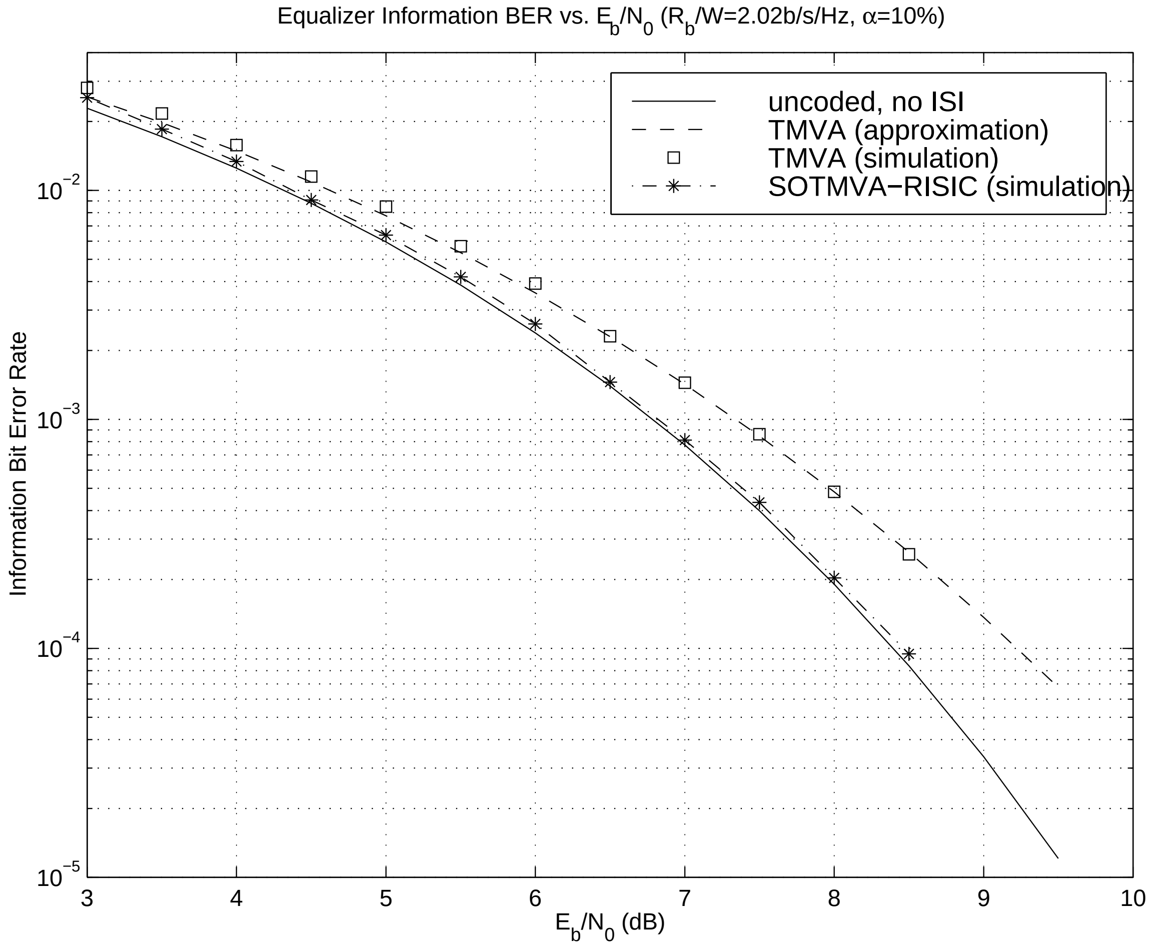

Fig. 59. Performance of the equalizer for FNS with $\tau = 9T/10$, simulated and approximated by (4.15) with $\rho = 0.0126x^2(0)$, compared to the optimum $Q(\sqrt{2E_b/N_0})$ (TMVA=Truncated Modified VA, SOTMVA=Soft-Output TMVA, RISIC=Residual ISI Canceller).

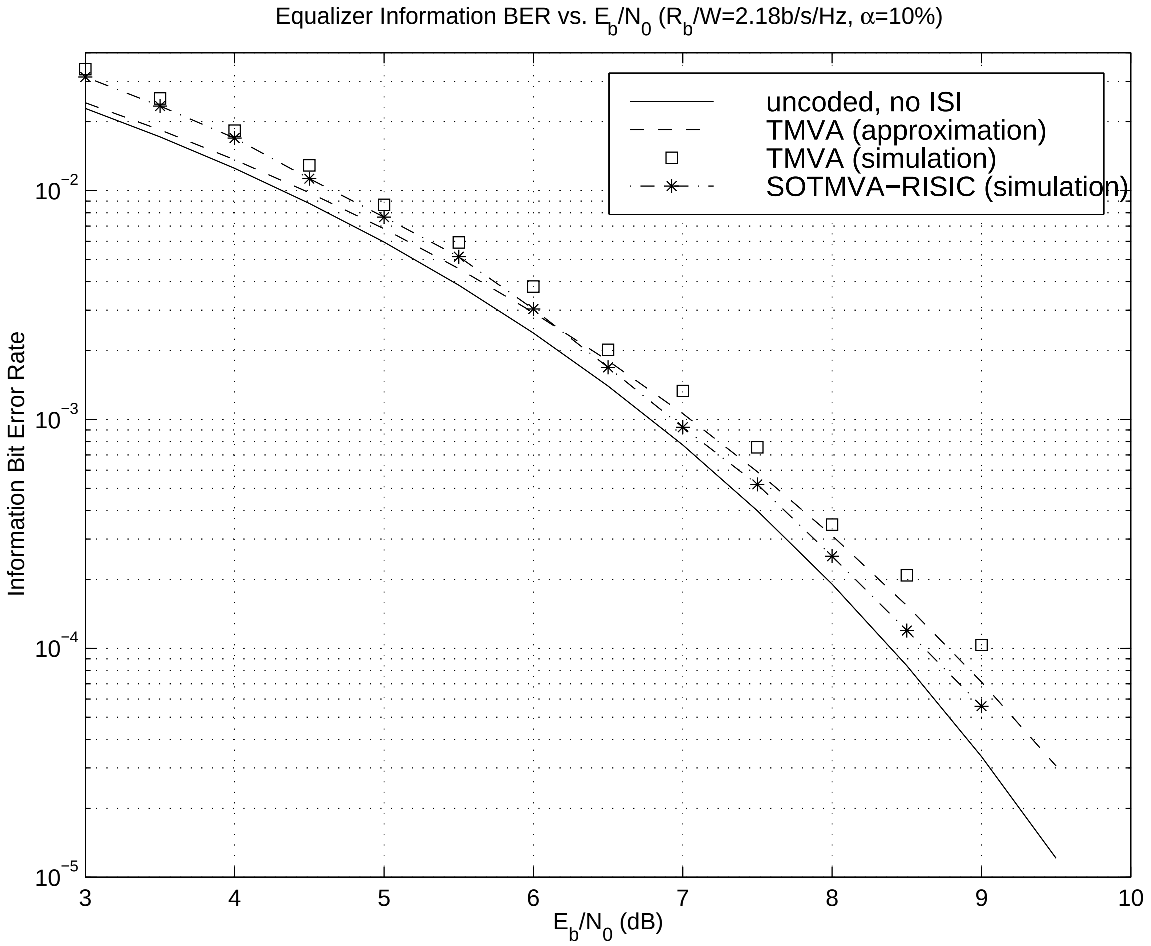

Fig. 60. Performance of the equalizer for FNS with $\tau = 5T/6$, simulated and approximated by (4.15) with $\rho = 0.0062x^2(0)$ compared to the optimum $Q(\sqrt{2E_b/N_0})$ (TMVA=Truncated Modified VA, SOTMVA=Soft-Output TMVA, RISIC=Residual ISI Canceller).

and the central tap), which are set to zero. To make the RISI estimate more reliable and, of course, to make the combining of the TMVA output with the RISI estimate possible, it is necessary that the TMVA provides soft-output values. The reason we expect that linear cancellation can improve the equalization results, is that the soft-output values in an optimum equalizer could be split into the sum of two terms: the first term is the scaled version of the input to the equalizer, the channel reliability, here $y_n 4E_b/N_0$, and the second is the extrinsic information of the equalizer, i.e., the "SNR-increase" it achieves by cancelling the ISI. In our suboptimum truncated detector, we make the assumption that the extrinsic information manages to cancel out the neighboring interference almost completely in the channel reliability term $y_n 4E_b/N_0$, so the linear filtering tries to cancel the residual ISI in this term.

Using the same approach as in [100], it is straightforward to modify the TMVA to a soft-output TMVA (SOTMVA) based on the TMVA trellis. If J_{1n} and J_{2n} are the metrics of the two merging paths at one state with $J_{1n} \geq J_{2n}$ (J_{1n} corresponds to the surviving path), which are computed from (4.9) at the n^{th} time instant, and $\Delta_n = J_{1n} - J_{2n} \geq 0$, the probability of selecting the wrong survivor path at the n^{th} time instant is [100]

$$p_{sn} = \frac{1}{1 + \exp(2\Delta_n E_b/N_0)} \leq 0.5 \tag{4.16}$$

where we took into account the proportionality factor from the derivation of (4.9) in [4] (exponent of the corresponding conditional Gaussian pdf). The error probabilities for the bits of the surviving path 1 (metric J_{1n}) are the same as before for all bits that are the same in paths 1 and 2,[4] and for the bits where the two paths differ, the

[4] We could further improve the soft-output values by considering the "corrected" SOVA [101].

updated bit error probabilities of the surviving path 1 are obtained from [100]

$$p_{jn} = p_{jn-1}(1 - p_{sn}) + (1 - p_{jn-1})p_{sn} \tag{4.17}$$

where j varies over all the positions path 1 and path 2 are different. Since p_{jn} is the probability of the j^{th} bit being in error at the n^{th} time instant, we consider $p_j = p_{jj+\delta}$ to be the output of our version of the SOVA, which is obtained with some delay $\delta = n - j$ (here we took $\delta = 6(L + 1)$ as in the case of convolutional codes). Then the corresponding log-likelihood ratio L_j is

$$L_j = \ln\left(\frac{1 - p_j}{p_j}\right) \tag{4.18}$$

and the corresponding j-th soft-output value is

$$\Lambda_j = \hat{a}_j L_j \tag{4.19}$$

where $\hat{a}_j \in \{-1, +1\}$ is the hard-output of the VA.

The way we can improve the RISI estimate is to provide to the filter input the expected values of the $\hat{a}_j$'s based on their reliability values $|\Lambda_j| = L_j$. Since from the definition of the log-likelihood ratio $\Lambda_j = \ln(P(\hat{a}_j = +1|\boldsymbol{y})/P(\hat{a}_j = -1|\boldsymbol{y})$, the expected value of $\hat{a}_j$ can be computed from

$$E[\hat{a}_j|\boldsymbol{y}] = \frac{e^{\Lambda_j} - 1}{e^{\Lambda_j} + 1}. \tag{4.20}$$

The proposed equalizer structure is shown in Fig. 61. We just note that the RISI estimate is first scaled by $4E_b/N_0$ and then subtracted from the delayed SOTMVA-output to yield the overall equalizer output, which with a sign detector would give the final bit decisions.

The scaling is required because of the way the matched filter output samples are related to the log-likelihood ratio. As already mentioned, at the output of the

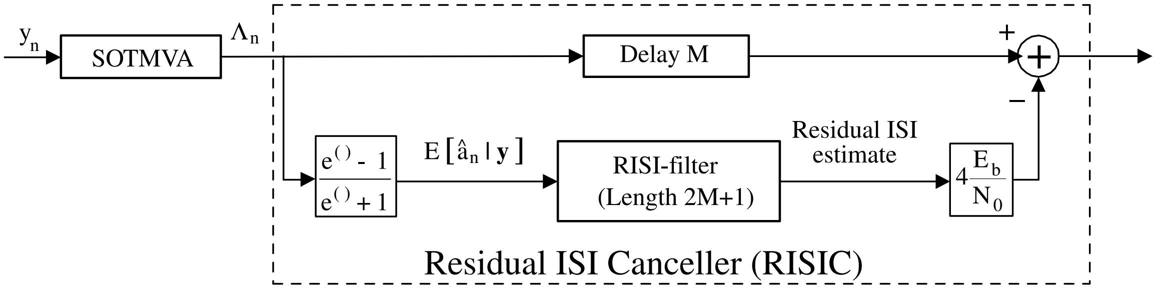

Fig. 61. Equalization scheme SOTMVA-RISIC with M being the one-sided length of the ISI, which is theoretically infinite but here we take $M = \lfloor 40T/\tau \rfloor$ (TMVA=Truncated Modified VA, SOTMVA=Soft-Output TMVA).

SOTMVA, the Λ_j's can approximately be split into two terms: the channel reliability $4y_nE_b/N_0$ and the extrinsic information resulting from the equalization of the $2L$-ISI taps. Since the RISIC unit cancels the rest of the ISI taps, the estimated RISI in y_n must be multiplied with $4E_b/N_0$ as well.

The performance of the equalizer with the RISI cancellation has been presented in Figs. 59 and 60, where we see that for both cases the two-stage scheme (SOTMVA-RISIC) performs better and quite close to the theoretically optimum ISI-free antipodal signaling performance for moderate to high SNRs. The results indicate that it is possible to signal at rates higher than Nyquist with almost no performance loss, at the expense of higher complexity and higher transmitted power (as in the case of multilevel modulation schemes).

E. Bandwidth Efficient Turbo Equalization

1. System Description

The bit-rate gains of the previous section can be partly or wholly translated into coding gains by using codes of suitable rate, i.e., code rates that result in information

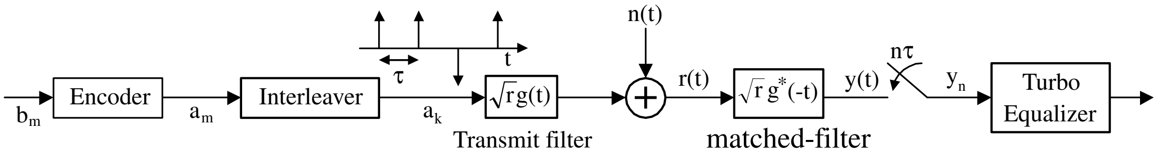

Fig. 62. Overall system for coded faster-than-Nyquist signaling.

spectral efficiencies that are at least as high as in the ISI free case. This means rates $r \in [q, 1]$, where the lower rate limit $q = \tau/T$ sacrifices all the excess bit-rate gain and the upper limit, 1, retains the full excess bit-rate (no coding). Code rate values between the above two limits result in trade-offs between bit-rate gains and coding gains. In what follows, we will only use $r = q$, i.e., we will trade-off all of the throughput gain for a coding gain.

The system we will use is shown in Fig. 62. The interleaver between the encoder and transmit-filter attempts to decorrelate the ISI error events so that the decorrelated events can more efficiently be handled by a quite weak (due to its high rate) code. The interleavers are block $N \times N$ interleavers where N is the codeword length of the code and the transmission is continuous, meaning that due to the long ISI, bits from neighboring $N \times N$ blocks are interfering. The turbo-equalizer unit in Fig. 62 refers to the combined iterative equalizer-decoder, analyzed in the following subsection. We only employ single parity-check codes which have flexible-rate and allow for low-complexity soft-input-soft-output (SISO) decoding. They are assumed to satisfy even parity in each codeword (even number of +1's).

The system uses the same transmit and receive filters as in Fig. 4 where now the a_k's are the coded symbols and the coded system is energy-penalized with the code rate $r = (N-1)/N$ for a fair comparison.

2. The Turbo Equalizer

The principle of turbo equalization is the same as turbo decoding where the inner decoder is substituted with an equalizer [97]. Both the equalizer and the decoder have to process soft-input and provide soft-output for better performance. The equalizer used is the soft-output truncated modified Viterbi algorithm (SOTMVA), presented earlier. Its output is deinterleaved and then decoded using a soft-input soft-output (SISO) decoder. Such a decoder for the single parity check codes works well with the simple low-complexity Max-Log-MAP rule described in [102]. The output λ_j of the Max-Log-MAP SISO decoder is [102]

$$\lambda_j = \begin{cases} s_j(|\Lambda_j| + \min_{k\neq j}|\Lambda_k|), & \text{even parity satisfied} \\ s_j(|\Lambda_j| - \min_{k\neq j}|\Lambda_k|), & \text{even parity not satisfied} \end{cases} \tag{4.21}$$

where Λ_j is the deinterleaved output of the SOTMVA, $s_j = \text{sign}(\Lambda_j)$, j, k vary over all N indices that correspond to a single codeword of length N and whether the even parity of the codeword under consideration is satisfied or not is checked by using just the signs of the coded bits s_j, $j = 1, 2, \ldots, N$. From (4.21) we see that the output λ_j of the SISO decoder is the sum of two terms: the input Λ_j and the extrinsic information $\lambda_j^e = \pm s_j \min_{k\neq j}|\Lambda_k|$. If we wanted to stop and not process the decoder output further, then the signs of the λ_j's for the K information bits in every codeword would be the most reliable estimate of the transmitted information bits[5]. But there is significant performance improvement if the extrinsic information together with some kind of RISI estimate are fed back to the SOTMVA and the detecting-decoding process is repeated at least once more. The residual ISI is better

[5] If the received sequence was passed just once through the detector-decoder, we could probably also try cancelling the RISI at the output of the decoder with linear filtering similarly to Subsection 2 in Section D.

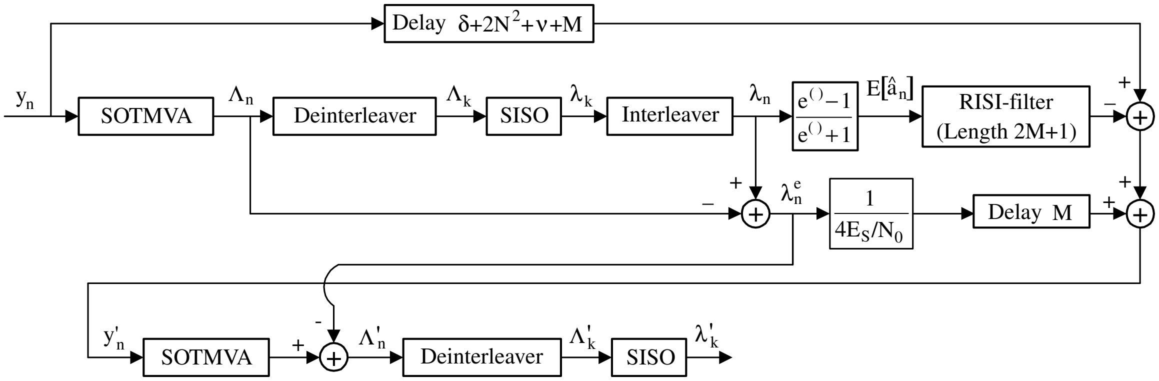

Fig. 63. Two-stage turbo equalizer for coded FNS (equivalent to a twice iterated one-stage turbo equalizer with feedback).

estimated at the feedback due to the higher reliability of the λ_j's compared to that of the Λ_j's.

The corresponding turbo equalizer structure is shown in Fig. 63, where the pipelined structure of two identical turbo equalizers instead of a single iterative one is depicted for better illustration of its continuous-time operation. The RISI-filter used is the same as the one discussed in Subsection 2 of Section D. The delay line for the matched filter output samples y_n's considers the delay δ of the SOTMVA, the $2N^2$ delay due to the deinterleaving-interleaving, the delay ν because of the SISO-decoder and the M-symbol delay of the RISI-filter. No scaling is needed this time for the RISI estimate, since it is subtracted directly from the matched-filter output.

The only thing still to be addressed is the extension of the SOTMVA, so that it also takes into account the a priori information, i.e., the decoder extrinsic information. This was done by Ungerboeck in Appendix II of [4]. Adjusting the proportionality factors to our case and expressing the a priori probabilities in terms of the extrinsic log-likelihood ratios, we get for the recursive relationship [4]

$$\begin{aligned} J_n(a_{n-L+1},\ldots,a_n) &= J_{n-1}(a_{n-L},\ldots,a_{n-1}) + a_n\left(y_n - \sum_{i=1}^{L} x(i\tau)\,a_{n-i}\right) \\ &\quad + \frac{N_0}{2E_s}\ln[P(a_n = j)], \end{aligned} \tag{4.22}$$

where $j = \pm 1$ and $E_s/N_0 = rE_b/N_0$ is the SNR per coded bit.

Since we are going to use estimates of $P(a_n = j)$ based on $\lambda_n^e \approx \ln(P(a_n = +1|decoding)/P(a_n = -1|decoding)$, $P(a_n = \pm 1)$ can be approximated by

$$P(a_n = -1) \approx \frac{1}{1 + e^{\lambda_n^e}}, \tag{4.23}$$

$$P(a_n = +1) \approx \frac{e^{\lambda_n^e}}{1 + e^{\lambda_n^e}}, \tag{4.24}$$

which can be written as

$$\ln[P(a_n = j)] \approx \frac{j+1}{2}\,\lambda_n^e - \ln(1 + e^{\lambda_n^e}), \quad j = \pm 1. \tag{4.25}$$

Dropping all common terms in (4.25) and substituting in (4.22), the recursive metric relationship becomes

$$\begin{aligned} J_n(a_{n-L+1},\ldots,a_n) = {} & J_{n-1}(a_{n-L},\ldots,a_{n-1}) \\ & + a_n\left(y_n + \frac{\lambda_n^e}{4E_s/N_0} - \sum_{i=1}^{L} x(i\tau)\,a_{n-i}\right), \end{aligned} \tag{4.26}$$

i.e., the scaled extrinsic information can be added to the matched-filter output samples directly, as shown in Fig. 63.

The scheme of Fig. 63 is equivalent to a single turbo-equalizer iterated twice. Its output information bit error rate, as well as that for one and more than two iterations are plotted in Figs. 64 and 65 for the single parity check codes with rates 9/10 and 5/6, respectively. Thus, the symbol spectral efficiencies, i.e., the coded bit rates, are respectively $R_s/W = 2.02$ bits/sec/Hz and $R_s/W = 2.18$ bits/sec/Hz, while the information spectral efficiency (information bit rate) is in both cases $R_b/W = 1.82$ bits/sec/Hz. In the plots of Figs. 64 and 65 the simulated performance of a coded system without ISI is presented (lower information rate for the same bandwidth) as is the analytical uncoded ISI-free performance. The first term of the union bound (smallest distance) for single parity check codes,

$$P(e) \approx (N-1)Q\left(\sqrt{\frac{4rE_b}{N_0}}\right), \tag{4.27}$$

is also plotted. In Fig. 64, because of the quite weak 9/10 parity check code, the number of iterations was limited to two since more iterations would not produce significant performance improvement. In any case, most of the gain of the second iteration even for the stronger 5/6 code results from the RISI cancellation and not from the extrinsic information. For both codes the gains are more than 1dB at a bit-error-rate of 10^{-4} and about 1.5dB at 10^{-5} for the 9/10 case compared to ISI-free transmission and for the same bandwidth and bit-rate. In other words, the increased signaling rate has been translated into a coding gain. The larger deviation of the 5/6 code from the optimum is a consequence of both the insufficient 6-by-6 block interleaving (not sufficient decorrelating of the long error events which start affecting performance) and the slightly worse performance of the suboptimum equalizer (Fig. 60) in this case. It is important to note that the turbo equalizer for both cases almost achieves the optimum ISI-free coded performance after two iterations for the same bandwidth, power and information rate as the uncoded ISI-free system (when the ISI-free system uses the same raised-cosine pulse shape).

We finally mention one example application of this turbo equalization scheme for $q < r < 1$. When one coded bit is transmitted every $\tau = 5T/6$, instead of using the 5/6 parity check code, we could use an 11/12 parity check code. For the rolloff

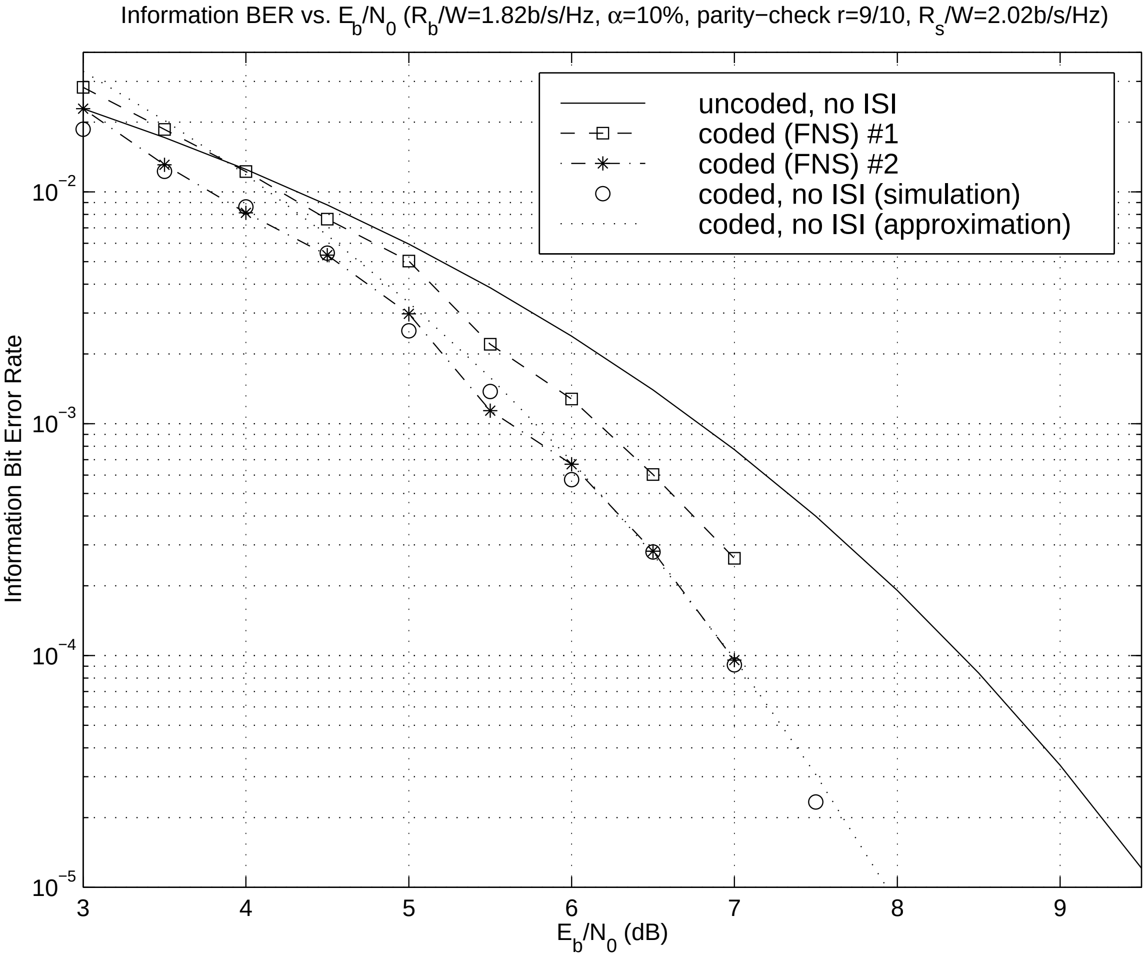

Fig. 64. Simulated performance of turbo equalizer for coded FNS (parity-check code 9/10, $\tau/T = 9/10$) after 1 and 2 iterations compared with the ISI free uncoded and ISI free coded systems (approximation refers to (4.27)).

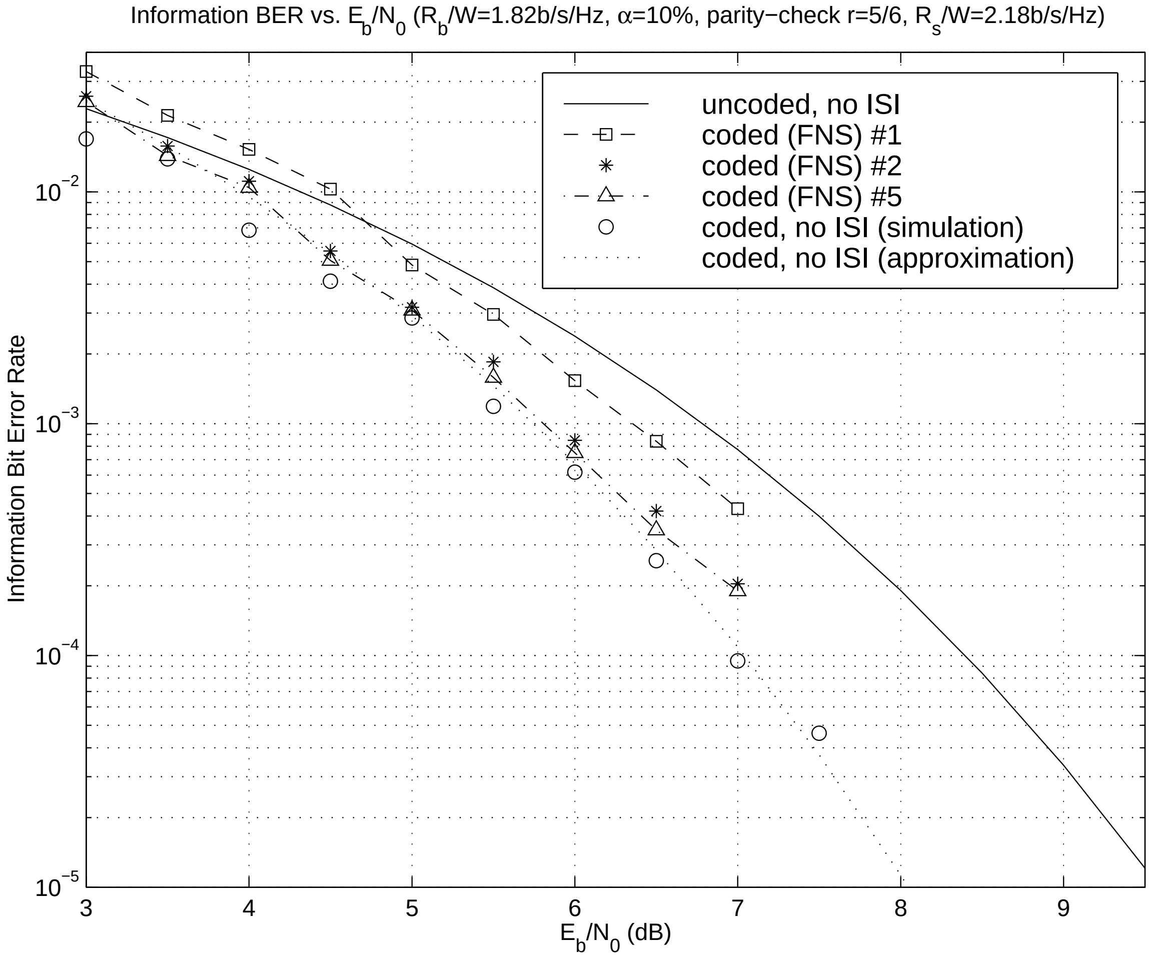

Fig. 65. Simulated performance of turbo equalizer for coded FNS (parity-check code 5/6, $\tau/T = 5/6$) after 1, 2 and 5 iterations compared with the ISI-free uncoded and ISI-free coded systems (approximation refers to (4.27)).

factor $\alpha = 10\%$, this corresponds to a signaling rate of $R_s/W = 2.18$b/s/Hz and an information rate of $R_b/W = 2.00$b/s/Hz, i.e. the excess bandwidth is accounted for by signaling faster and at the same time we expect to have some coding gain due to the 12-by-12 block interleaving. With this simple coding scheme only part of the higher signaling rate could be translated into coding gain and still communicate at a higher information rate. All these practical gains could possibly be extended by employing more powerful codes, like Hamming codes or punctured convolutional codes with precoding, at the price of increased decoding complexity. But, of course, the question of what the ultimate limits of FNS are, will still be unanswered.

F. Bandwidth and Achievable Throughput

In this section we try to determine the throughput of faster-than-Nyquist signaling (FNS) with the help of the power spectral density (PSD) of the transmitted signal. The whole analysis is based on the ideal sinc pulses ("brickwall" spectrum). It turns out that for i.i.d. zero-mean Gaussian input, signaling faster is equivalent to increasing the average signaling power. In what follows, we consider the ideal sinc pulses of bandwidth W

$$g(t) = \sqrt{2WE_s}\,\mathrm{sinc(2Wt)} = \sqrt{2WE_s}\,\frac{\sin(2\pi Wt)}{2\pi Wt} \tag{4.28}$$

which are normalized to energy E_s. The Fourier transform of $g(t)$ is

$$G(f) = \frac{\sqrt{E_s}}{\sqrt{2W}}\,\Pi\left(\frac{f}{2W}\right) = \begin{cases} \frac{\sqrt{E_s}}{\sqrt{2W}}, & |f| < W, \\ \\ 0, & |f| > W. \end{cases} \tag{4.29}$$

The Nyquist signaling rate for these pulses is $2W$ and in this case is the same as the ISI-free rate (no roll-off). Assuming baseband transmission, the transmitted signal

$v(t)$ has the form

$$v(t) = \sum_{n=-\infty}^{\infty} a_n \, g(t - n\tau) \tag{4.30}$$

where throughout this section the input symbols a_n are assumed (real) zero-mean, independent and identically distributed (i.i.d.) with unit variance , i.e., $E[a_{n+m}a_n] = \delta_m$. Signaling faster-than-Nyquist (FNS) in this context means that $1/\tau > 2W$. The most important result in this section is that, for Gaussian input, the effect of FNS on the achievable information rates is equivalent to increasing the transmitted average power $P = E_s/\tau$ by reducing τ while keeping E_s fixed, i.e., it is an alternative to increasing the average symbol energy E_s and keeping τ fixed. In the following subsection we will reach some more precise conclusions about the power spectral density (PSD) of $v(t)$ starting from the general ones of [103]. Then, based on these results, we will try to determine the information rates achieved by this FNS scheme.

1. Power Spectral Density of Transmitted Signal

According to [103], the process $v(t)$ is in general cyclostationary, since its autocorrelation function

$$R_{vv}(t+u,t) \;\; = \;\; E[v(t+u)\, v(t)] = \sum_{n=-\infty}^{\infty} g(t+u-n\tau)\, g(t-n\tau), \tag{4.31}$$

is periodic in t with period τ. Due to its periodic nature, a Fourier series expansion of $R_{vv}(t+u,t)$ in t is possible

$$R_{vv}(t+u,t) = \sum_{k=-\infty}^{\infty} R_k(u)\; \mathrm{e}^{j2\pi \frac{k}{\tau} t}, \tag{4.32}$$

where $R_k(u)$ can be determined from

$$R_k(u) = \frac{1}{\tau} \int_{-\tau/2}^{\tau/2} R_{vv}(t+u,t)\; \mathrm{e}^{-j2\pi \frac{k}{\tau} t}\, dt\,. \tag{4.33}$$

Using (4.31) and changing the integration variable we obtain

$$\begin{aligned} R_k(u) &= \frac{1}{\tau} \sum_{n=-\infty}^{\infty} \int_{n\tau-\tau/2}^{n\tau+\tau/2} g(t+u)\, g(t)\ \mathrm{e}^{-j2\pi \frac{k}{\tau} t}\, dt \\ &= \frac{1}{\tau} \int_{-\infty}^{\infty} g(t+u)\, g(t)\ \mathrm{e}^{-j2\pi \frac{k}{\tau} t}\, dt \\ &= \frac{1}{\tau} \left[G(f)\, \mathrm{e}^{j2\pi f u} * G(f) \right]_{f=k/\tau} \\ &= \frac{E_s}{2W\tau} \int_{-W}^{W} \Pi\left(\frac{\frac{k}{\tau} - x}{2W} \right)\ \mathrm{e}^{j2\pi u x}\, dx. \end{aligned} \tag{4.34}$$

where in the last step we used (4.29). From the last expression for $R_k(u)$, it is clear that for $1/\tau \geq 2W$, i.e., for rates greater than or equal to the Nyquist rate, $R_k(u) = 0$ for all $k \neq 0$. For $k = 0$

$$R_0(u) = \frac{E_s}{\tau} \operatorname{sinc}(2Wu). \tag{4.35}$$

Thus, for sinc pulses, for zero-mean i.i.d. input and for signaling at least at the Nyquist rate, the autocorrelation of $v(t)$ is actually a function of u alone, i.e., $v(t)$ is a wide-sense stationary (WSS) process. (It is straightforward to generalize this result to any strictly bandlimited pulse shape to W when signaling at $1/\tau > 2W$.) The autocorrelation and PSD of this WSS process are respectively

$$R_{vv}(\tau) = \frac{E_s}{\tau} \operatorname{sinc}(2Wu), \tag{4.36}$$

$$S_{vv}(f) = \frac{E_s}{2W\tau} \Pi\left(\frac{f}{2W} \right), \tag{4.37}$$

i.e., they are equal to the average autocorrelation and PSD when the cyclostationary approach of [103] is used. However, the actual autocorrelation and PSD are more directly linked to the statistics of the signal and can help us reach more conclusions as will be shown next.

2. Throughput of FNS

The surprising fact about the result of the previous section is the fact that sampling the transmitted process at the Nyquist rate $2W$, the samples are uncorrelated as (4.36) shows. Assuming an additive white Gaussian noise (AWGN) channel and a brickwall (matched) filter $g(t)$, this property still holds at the matched filter output. The PSD at the matched filter output is

$$S_{yy}(f) = \left(\frac{E_s}{2W\tau} + \frac{N_0}{2}\right) \frac{E_s}{2W}\, \Pi\left(\frac{f}{2W}\right), \tag{4.38}$$

and so the variance σ_y^2 of the Nyquist rate samples, i.e., at time instants $n/2W$, $n \in Z$, is

$$\sigma_y^2 = \int_{-\infty}^{\infty} S_{yy}(f)\, df = \frac{E_s^2}{2W\tau} + \frac{N_0 E_s}{2}. \tag{4.39}$$

At the same time these samples are sufficient statistics of the received signal, since it is strictly bandlimited to W Hz. In Section G, we will show how these samples can be used to derive the same log-likelihood function as in (4.8) for the sinc pulses and for a discrete input alphabet.

However, for the rest of this section we will consider Gaussian input symbols a_n in (4.30). Then the matched filter output Nyquist-rate samples are also Gaussian, as a linear combination of Gaussian random variables, and because they are uncorrelated, they are independent. For this case it is easy to determine the corresponding information rates of this FNS scheme. Starting from $I(\boldsymbol{X};\boldsymbol{Y}) = h(\boldsymbol{Y}) - h(\boldsymbol{Y}|\boldsymbol{X})$, where $\boldsymbol{X}$ represents the sequence of the information (input) symbols (length N), $\boldsymbol{Y}$ the sequence of the matched filter output samples (length $(2W\tau)N \leq N$) and $I(\cdot;\cdot)$, $h(\cdot)$ refer to the mutual information and differential entropy respectively, we get due to the independence of the components of $\boldsymbol{Y}$ and their associated noise parts

$Z_i \sim N(0, E_s N_0/2)$

$$\begin{aligned} I(\boldsymbol{X};\boldsymbol{Y}) &= \sum_{i=0}^{(2W\tau)N-1} h(Y_i) - \sum_{i=0}^{(2W\tau)N-1} h(Z_i) \\ &= NW\tau \, \log_2\left(\frac{\sigma_y^2}{E_s N_0/2}\right) \\ &= NW\tau \, \log_2\left(1+\frac{E_s}{W\tau N_0}\right), \end{aligned} \tag{4.40}$$

where in the last step we substituted from (4.39). Hence, the information rate in bits per transmitted symbol is $W\tau \, \log_2(1 + E_s/(W\tau N_0))$, whereas in bits per matched filter output sample it is $\log_2(1+E_s/(W\tau N_0))/2$. The throughput in bits per seconds is

$$C_{FNS} = W \, \log_2\left(1+\frac{E_s}{W\tau N_0}\right), \tag{4.41}$$

i.e., the effect of FNS in this system is like an alternative way of increasing the transmitted power by signaling faster instead of increasing the symbol energy. In other words, the importance of this result lies in the fact that there is no loss in the best possible bandwidth efficiency when employing an FNS scheme. Of course, this is only a small step towards the final result we wish to obtain, which is to compare the achievable throughput of practical constellation schemes with and without FNS as well as the design of receivers able to operate with matched filter output samples taken at the Nyquist rate.

G. Sampling at Constant Rate

In this section we try to determine some possible simplifications in implementing faster-than-Nyquist signaling (FNS), focusing more on the sampling rate at the receiver. The whole analysis is based on the ideal sinc pulses (“brickwall” bandwidth). It turns out that for the usual binary i.i.d. input FNS system, sampling faster at the receiver, i.e., at a rate equal to the signaling rate, is equivalent to sampling at exactly the Nyquist rate. The equivalence is asymptotical and is established by getting the same expression both for the Euclidean distance of two distinct data sequences and for the metric used in the Viterbi algorithm of the optimum detector, no matter which of the two sampling rates was used.

1. Log-Likelihood Function at the Nyquist Sampling Rate

The Nyquist-rate samples $y\left(\frac{n}{2W}\right)$ are sufficient statistics of the received signal, as shown in Subsection 2 of Section F, and at the same time their noise components are independent with variance $E_s \frac{N_0}{2}$. Apart from the practical aspects of this sampling rate, like keeping the constant sampling rate in a variable signaling rate system, accommodating the varying rate by adapting the system coefficients, due to the white noise samples it is simpler to express the conditional joint probability density function (pdf) in the discrete time domain. This conditional joint pdf of the matched-filter output samples $y_n = y\left(\frac{n}{2W}\right)$ which is

$$p(\boldsymbol{y}|\boldsymbol{a}) = \prod_{n=-N}^{N} p(y_n|\boldsymbol{a}) = \frac{1}{(\sqrt{\pi N_0 E_s})^{2N+1}} \exp\left(-\frac{\sum_{n=-N}^{N}(y_n - f_n(\boldsymbol{a}))^2}{N_0 E_s}\right), \quad (4.42)$$

where the number of output samples $(2N+1)$ is very large, we will consider it to be infinite from now on, and

$$f_n(\boldsymbol{a}) = \sum_{j=-N/(2W\tau)}^{N/(2W\tau)} h_{n,j}\, a_j, \quad (4.43)$$

with $h_{n,j} = E_s \operatorname{sinc}(n - j2W\tau)$ and $1/\tau > 2W$ being the faster rate. The input symbols are assumed to be zero-mean i.i.d. and to belong to a discrete alphabet.

As we see from (4.42) and (4.43), there are two side effects: first of all, the interference pattern is different from sample to sample (periodically repeating if $2W\tau$

is rational) and there are less samples than input symbols (for every $N/(2W\tau)$ input bits, we get N matched-filter output samples).

Maximizing the pdf of (4.42) is equivalent to maximizing

$$\ell'(\boldsymbol{a}) = \sum_n \left[y_n\, f_n(\boldsymbol{a}) - \frac{1}{2}\, f_n^2(\boldsymbol{a}) \right], \tag{4.44}$$

over all possible $\boldsymbol{a}$, with $f_n(\boldsymbol{a})$ as in (4.43). Substituting from (4.43), the last expression becomes

$$\begin{aligned} \ell'(\boldsymbol{a}) &= \sum_n \sum_j \left[y_n\, h_{n,j}\, a_j - \frac{1}{2}\, h_{n,j}\, a_j \sum_k h_{n,k}\, a_k \right] \\ &= \sum_j a_j \sum_n y_n\, h_{n,j} - \frac{1}{2} \sum_j \sum_k a_j\, a_k \sum_n h_{n,j}\, h_{n,k}. \end{aligned} \tag{4.45}$$

Using properties of the sinc function, it can be proven that $\sum_n h_{n,i}\, h_{n,k} = E_s\, h_{0,i-k}$ and that $\sum_n y_n\, h_{n,j} = E_s\, y(j\tau)$. For the former, we take the discrete time Fourier transform (DTFT) of $\text{sinc}(n-\theta)$, i.e.

$$D_\theta(\omega) = \mathcal{F}\left\{\text{sinc}(n-\theta)\right\} = \mathrm{e}^{-j\omega\theta}\, \Pi\left(\frac{\omega}{2\pi}\right), \tag{4.46}$$

and then apply Parseval's theorem for the inner product of the left hand side

$$\begin{aligned} \sum_n h_{n,i}\, h_{n,k} &= E_s^2 \sum_n \text{sinc}\,(n - i2WT)\ \text{sinc}\,(n - k2WT) \\ &= E_s^2\, \frac{1}{2\pi} \int_{-\pi}^{\pi} D_{i2WT}(\omega)\, D_{k2WT}^*(\omega)\, d\omega \\ &= E_s^2\, \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathrm{e}^{j\omega(k-i)2WT}\, d\omega \\ &= E_s^2\, \text{sinc}\,((i-k)2WT) = E_s\, h_{0,i-k}. \end{aligned} \tag{4.47}$$

As for $\sum_n y_n\, h_{n,j} = E_s\, y(j\tau)$, the left hand side is the interpolation formula for the reconstruction of $y(t)$ at the time instants $t = j\tau$ with sinc functions and its samples at $t = \frac{n}{2W}$. So the equivalent expression to be maximized becomes

$$\ell'(\boldsymbol{a}) = E_s \sum_j a_j \, y(j\tau) - \frac{E_s}{2} \sum_j \sum_k a_j \, a_{j-k} \, h_{0,k}, \tag{4.48}$$

which is equivalent to (4.8) (the difference in the multiplicative factors can be cancelled out).

However, in the proofs we used two key relationships, $\sum_n h_{n,i} \, h_{n,k} = E_s \, h_{0,i-k}$ and $\sum_n y_n h_{n,j} = E_s \, y(j\tau)$. Although both assume that we operate on infinite sequences, the latter one really causes a practical problem as it assumes infinite delay. Therefore other means of interpolation might be more desirable, yielding an estimate $\hat{y}(j\tau)$ of $y(j\tau)$. The small associated loss is the prize paid for having a system able to operate at variable signaling rates.

Based on the same assumptions, a similar conclusion can be drawn about the Euclidean distance $d(\boldsymbol{a}, \boldsymbol{b})$ between two sequences $\boldsymbol{a}$ and $\boldsymbol{b}$, i.e., regardless whether the output of the matched-filter is sampled at the constant Nyquist rate $2W$, or the FNS rate $\frac{1}{\tau} \geq 2W$, $d(\boldsymbol{a}, \boldsymbol{b})$ remains the same.

H. Conclusions and Discussion

From all the presented results, we can conclude that for ideally bandlimited systems with constrained average power, FNS appears to be a way of trading off performance (smaller E_s) to higher information rates (smaller τ) practically without any loss. Most of this chapter had to do with methods of exploiting the more practical gains that come up from the minimum distance estimation of the binary FNS system. First, with the use of constrained coding, ways of extending the theoretical d_{min} gains to even higher signaling rates were suggested. Then using soft interference cancellation techniques combined with turbo equalization, we can realize some of the theoretically predicted gains. In particular, our results using raised-cosine pulses with small roll-off, indicate practically implementable throughput gains and/or performance gains by using coding. We also ended up with an interesting recursive metric for the detector operating with Nyquist rate samples, but we did not implement such a scheme.

Some interesting extensions of the reported results are the use of more powerful codes to achieve better trade-offs of throughput gains for coding gains, the optimization within the class of raised-cosine signaling pulses with respect to the system parameters (roll-off α, length of ISI L in the TMVA, ratio τ/T) and the FNS-oriented study and design of pulses.

Even more promising gains from FNS could probably be obtained by employing some form of precoding, like faster partial response signaling [104, 105] or FNS oriented precoding [106, 107, 108]. Another related extension of FNS can involve $M-$ary signaling with $M > 2$, where with some increased complexity in the analysis and the design, some FNS gains are expected with the same receiver structures. However, the nonbinary case [108] has not been studied as well as the binary case in the literature, leaving the question of the corresponding FNS gains open.

An interesting open issue is how binary or nonbinary FNS compares to other bandwidth efficient schemes like nonbinary modulation at the Nyquist rate. Associated results in the literature exist only for the uncoded case [89, 90, 91] and are slightly in favor of Nyquist-rate systems. But even if the comparison was clearly in favor of the Nyquist-rate system in both the uncoded and the coded case, FNS schemes would still find applications in scenarios where higher bandwidth power efficiency is desirable without the possibility to expand the constellation.

A last significant issue, throughput analysis, i.e., determining the achievable information rate for FNS, with practical input symbols, like binary or $M-$ary, has still not been addressed.

CHAPTER V

SUMMARY

This dissertation included three parts: distributed coding, quantization of coded channel measurements, and faster-than-Nyquist signaling. The contribution of each of these parts is summarized in the following three sections.

A. Distributed Coding

In the first part of this dissertation we studied three distributed coding problems. The first was Slepian-Wolf coding, including both asymmetric and symmetric coding, as well as coding of multiple sources. All the proposed Slepian-Wolf codes converted the Slepian-Wolf coding problem, i.e., a lossless distributed source coding problem, into an equivalent channel coding problem. So the use of good classical channel codes guaranteed good performance for the corresponding Slepian-Wolf codes. Thus, exploiting the progress in the mature field of classical channel code design, the proposed Slepian-Wolf coding schemes in this dissertation could come very close to the theoretical limits.

The second distributed coding problem in this part of the dissertation was asymmetric Slepian-Wolf coding over a noisy channel. This source-channel coding problem can be converted to an equivalent systematic channel coding problem over two parallel channels, i.e., it has a channel coding equivalent which is not conventional. Taking into account the two parallel channels in the channel code design however, enables the design of good source-channel coding schemes for this problem.

The last distributed coding problem considered in this dissertation was Wyner-Ziv coding. This source coding problem could be converted to a source-channel coding problem and several approaches to code design where presented and analyzed, leading to the one with the best performance. No other results have been reported in the literature for the binary Wyner-Ziv problem considered here.

B. Quantization of Coded Channel Measurements

The second part of this dissertation studied several quantizer design approaches for quantization of the sampled received values at the receiver in a coded point-to-point communication system. The main contribution of this part of the dissertation has been the extensive comparison of the quantization criteria for uniform and nonuniform scalar quantization. This comparison establishes the advantage of cutoff rate and mutual information over MSE and this advantage is further confirmed through simulations of powerful LDPC codes.

Two interesting results of the comparison between the different quantizers are the negligible gains of nonuniform quantization over uniform and the strong robustness of good quantizers, i.e., by appropriately choosing a quantizer it can be used over a wide range of SNRs without noticeable performance loss. Both these results are again confirmed by simulations.

C. Faster-than-Nyquist Signaling

The third part of this dissertation studied several ways of signaling faster than the Nyquist rate without suffering any loss in the performance due to the higher signaling rate and the induced intersymbol interference. The interference was mitigated through reduced complexity equalization combined with constrained coding when it was severe or just reduced complexity equalization when it was less severe.

The bit rate gains that were achieved were also translated into coding gains by employing channel coding and turbo equalization. For the first time it was exhibited that faster-than-Nyquist signaling has some potential as an alternative signaling scheme.

REFERENCES

[1] J. Garcia-Frias and Y. Zhao, “Compression of correlated binary sources using turbo codes,” *IEEE Commun. Letters*, vol. 5, pp. 417–419, Oct. 2001.

[2] J. Garcia-Frias and Y. Zhao, “Compression of binary memoryless sources using punctured turbo codes,” *IEEE Commun. Letters*, vol. 6, pp. 394–396, Sept. 2002.

[3] A. Aaron and B. Girod, “Compression with side information using turbo codes,” in *Proc. DCC ’02*, Apr. 2002, pp. 252–261.

[4] G. Ungerboeck, “Adaptive maximum-likelihood receiver for carrier-modulated data-transmission systems,” *IEEE Trans. Commun.*, vol. COM-22, pp. 624–636, May 1974.

[5] P.J. McLane, “A residual interference error bound for truncated state detectors,” *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 548–553, Sept. 1980.

[6] A. Hafeez and W.E. Stark, “Decision feedback sequence estimation for unwhitened ISI channels with applications to multiuser detection,” *IEEE J. Select. Areas Commun.*, vol. JSAC-16, pp. 1785–1795, Dec. 1998.

[7] S.S. Pradhan and K. Ramchandran, “Distributed source coding using syndromes (DISCUS): Design and construction,” in *Proc. DCC ’99*, Mar. 1999, pp. 158–167.

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

[9] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* New York: John Wiley & Sons, 1991.

[10] A. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. ICC '93*, May 1993, pp. 1064–1070.

[11] R.G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[12] D.J.C. MacKay and R.M. Neal, "Near Shannon limit performance of low-density parity-check codes," *Electron. Letters*, vol. 33, pp. 457–458, Mar. 1997.

[13] M. Sipser and D.A. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 1710–1722, Nov. 1996.

[14] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," in *Proc. GLOBECOM '01*, Nov. 2001, vol. 2, pp. 1400–1404.

[15] T. Murayama, "Statistical mechanics of the data compression theorem," *J. Phys. A: Math. Gen.*, vol. 35, pp. L95–L100, Mar. 2002.

[16] A.D. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 2–10, Jan. 1974.

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

[18] S.-Y. Chung, T.J. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. IT-47, pp. 657–670, Feb. 2001.

[19] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. IT-47, pp. 619–637, Feb. 2001.

[20] S.-Y. Chung, "On the Construction of Some Capacity-Approaching Coding Schemes," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[21] S.-Y. Chung, G.D. Forney, 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. Letters*, vol. 5, pp. 58–60, Feb. 2001.

[22] G.D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.

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

[24] K.E. Tepe and J.B. Anderson, "Turbo codes for binary symmetric and binary erasure channels," in *Proc. ISIT '98*, Aug. 1998, p. 59.

[25] O.F. Acikel and W.E. Ryan, "Punctured turbo-codes for BPSK/QPSK channels," *IEEE Trans. Commun.*, vol. COM-47, pp. 1315–1323, Sept. 1999.

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

[27] S.S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Trans. Inform. Theory*, vol. IT-49, pp. 626–643, Mar. 2003.

[28] J.K. Wolf, "Data reduction for multiple correlated sources," in *Proc. 5th Colloquium Microwave Communication*, June 1974, pp. 287–295.

[29] S.S. Pradhan and K. Ramchandran, "Generalized coset codes for distributed source coding," submitted to IEEE Trans. Inform. Theory, Feb. 2003.

[30] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Int. Symp. Turbo Codes and Rel. Topics*, Sept. 2000, pp. 1–8.

[31] N. Gehrig and P.L. Dragotti, "Symmetric and a-symmetric Slepian-Wolf codes with systematic and non-systematic linear codes," *IEEE Commun. Letters*, 2004, to appear.

[32] V. Stanković, A.D. Liveris, Z. Xiong, and C.N. Georghiades, "Design of Slepian-Wolf codes by channel code partitioning," in *Proc. DCC '04*, Mar. 2004, pp. 302–311.

[33] T. Cover, "A proof of the data compression theorem of Slepian and Wolf for ergodic sources (corresp.)," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 226–228, May 1975.

[34] A.D. Liveris, C.-F. Lan, K.R. Narayanan, Z. Xiong, and C.N. Georghiades, "Slepian-Wolf coding of three sources using LDPC codes," in *Proc. 3rd Int. Symp. Turbo Codes and Rel. Topics*, Sept. 2003, pp. 63–66.

[35] C.-F. Lan, A.D. Liveris, K.R. Narayanan, Z. Xiong, and C.N. Georghiades, "Slepian-Wolf coding of multiple M-ary sources using LDPC codes," in *Proc. DCC '04*, Mar. 2004, p. 549.

[36] D. Schonberg, K. Ramchandran, and S.S. Pradhan, "Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources,"

in *Proc. DCC '04*, Mar. 2004, pp. 292–301.

[37] T.P. Coleman, A.H. Lee, M. Medard, and M. Effros, "On some new approaches to practical Slepian-Wolf compression inspired by channel coding," in *Proc. DCC '04*, Mar. 2004, pp. 282–291.

[38] J. Garcia-Frias, "Joint source-channel decoding of correlated sources over noisy channels," in *Proc. DCC '01*, Mar. 2001, pp. 283–292.

[39] P. Mitran and J. Bajcsy, "Turbo source coding: A noise-robust approach to data compression," in *Proc. DCC '02*, Apr. 2002, p. 465.

[40] G.-C. Zhu and F. Alajaji, "Turbo codes for nonuniform memoryless sources over noisy channels," *IEEE Commun. Letters*, vol. 6, pp. 64–66, Feb. 2002.

[41] G.-C. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, "Non-systematic turbo codes for non-uniform i.i.d. sources over AWGN channels," in *Proc. CISS '02*, Mar. 2002, pp. 770–774.

[42] G.-C. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, "Transmission of non-uniform memoryless sources over wireless fading channels via non-systematic turbo codes," in *Proc. IASTED WOC '02*, July 2002, pp. 119–124.

[43] J. Garcia-Frias, W. Zhong, and Y. Zhao, "Iterative decoding schemes for source and joint source-channel coding of correlated sources," in *Proc. Asilomar '02*, Nov. 2002, vol. 1, pp. 250–256.

[44] W. Zhong, H. Lou, and J. Garcia-Frias, "LDGM codes for joint source-channel coding of correlated sources," in *Proc. ICIP '03*, Sept. 2003, vol. 1, pp. 593–596.

[45] C. Lan, K.R. Narayanan, and Z. Xiong, "Scalable image transmission using rate-compatible irregular repeat accumulate (IRA) codes," in *Proc. ICIP '02*, June 2002, vol. 3, pp. 717–720.

[46] S. Shamai (Shitz) and S. Verdú, "Capacity of channels with uncoded side information," *European Trans. Telecommun.*, vol. 6, pp. 587–600, Sept./Oct. 1995.

[47] H. Jin, "Analysis and Design of Turbo-like Codes," Ph.D. dissertation, California Institute of Technology, 2001, Chapter 7.

[48] S. Shamai, S. Verdú, and R. Zamir, "Systematic lossy source/channel coding," *IEEE Trans. Inform. Theory*, vol. IT-44, pp. 564–579, Mar. 1998.

[49] J. Chou, S.S. Pradhan, and K. Ramchandran, "Turbo and trellis-based constructions for source coding with side information," in *Proc. DCC '03*, Mar. 2003, pp. 33–42.

[50] A.R. Calderbank, P.C. Fishburn, and A. Rabinovich, "Covering properties of convolutional codes and associated lattices," *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 732–746, May 1995.

[51] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE J. Select. Areas Commun.*, vol. JSAC-19, pp. 891–907, May 2001.

[52] D. Schonberg, K. Ramchandran, and S.S. Pradhan, "LDPC codes can approach the Slepian-Wolf bound for general binary sources," in *Proc. Allerton '02*, Oct. 2002, pp. 576–585.

[53] J. Li, Z. Tu, and R.S. Blum, “Slepian-Wolf coding for nonuniform sources using turbo codes,” in *Proc. DCC ’04*, Mar. 2004, pp. 312–321.

[54] P. Mitran and J. Bajcsy, “Coding for the Wyner-Ziv problem with turbo-like codes,” in *Proc. ISIT ’02*, June/July 2002, p. 91.

[55] Y. Yang, S. Cheng, Z. Xiong, and W. Zhao, “Wyner-Ziv coding based on TCQ and LDPC codes,” in *Proc. Asilomar ’03*, Nov. 2003, vol. 1, pp. 825–829.

[56] Z. Liu, S. Cheng, A.D. Liveris, and Z. Xiong, “Slepian-Wolf coded nested quantization (SWC-NQ) for Wyner-Ziv coding: performance analysis and code design,” in *Proc. DCC ’04*, Mar. 2004, pp. 322–331.

[57] X. Wang and M.T. Orchard, “On reducing the rate of retransmission in time-varying channels,” *IEEE Trans. Commun.*, vol. COM-51, pp. 900 – 910, June 2003.

[58] W. Zhong, Y. Zhao, and J. Garcia-Frias, “Turbo-like codes for distributed joint source-channel coding of correlated senders in multiple access channels,” in *Proc. Asilomar ’03*, Nov. 2003, vol. 1, pp. 840–844.

[59] A. Aaron, S. Rane, R. Zhang, and B. Girod, “Wyner-Ziv coding for video: applications to compression and error resilience,” in *Proc. DCC ’03*, Mar. 2003, pp. 93–102.

[60] R. Puri and K. Ramchandran, “PRISM: a “reversed” multimedia coding paradigm,” in *Proc. ICIP ’03*, Sept. 2003, vol. 1, pp. 617–620.

[61] A. Sehgal, A. Jagmohan, and N. Ahuja, “Wyner-Ziv coding of video: an error-resilient compression framework,” *IEEE Trans. Multimedia*, vol. 6, pp. 249–258, Apr. 2004.

[62] Q. Xu and Z. Xiong, "Layered Wyner-Ziv video coding," in *Proc. VCIP '04*, Jan. 2004, pp. 83–91.

[63] G. Jeong and D. Hsia, "Optimal quantization for soft-decision turbo decoder," in *Proc. VTC '99*, Sept. 1999, vol. 3, pp. 1620–1624.

[64] U. Dasgupta and C.N. Georghiades, "Turbo decoding of quantized data," *IEEE Trans. Commun.*, vol. COM-50, pp. 56–64, Jan. 2002.

[65] Y. Wu and B.D. Woerner, "The influence of quantization and fixed point arithmetic upon the bit-error performance of turbo codes," in *Proc. VTC '99*, May 1999, vol. 2, pp. 1683–1687.

[66] J. Au and P.J. McLane, "Performance of turbo codes with quantized channel measurements," in *Proc. GLOBECOM '99*, Dec. 1999, vol. 5, pp. 2586–2591.

[67] H. Michel, A. Worm, and N. Wehn, "Influence of quantization on the bit-error performance of turbo-decoders," in *Proc. VTC '00*, May 2000, vol. 1, pp. 581–585.

[68] H. Michel and N. Wehn, "Turbo-decoder quantization for UMTS," *IEEE Commun. Letters*, vol. 5, pp. 55–57, Feb. 2001.

[69] G. Montorsi and S. Benedetto, "Design of fixed-point iterative decoders for concatenated codes with interleavers," *IEEE J. Select. Areas Commun.*, vol. JSAC-19, pp. 871–882, May 2001.

[70] L. Ping and W.K. Leung, "Decoding low density parity check codes with finite quantization bits," *IEEE Commun. Letters*, vol. 4, pp. 62–64, Feb. 2000.

[71] T. Zhang, Z. Wang, and K.K. Parhi, "On finite precision implementation of low density parity check codes decoder," in *Proc. ISCAS '01*, May 2001, vol. 4, pp. 202–205.

[72] J.A. Heller and I.M. Jacobs, "Viterbi decoding for satellite and space communications," *IEEE Trans. Commun.*, vol. COM-19, pp. 835–848, Oct. 1971.

[73] I.M. Onyszchuk, K.-M. Cheung, and O. Collins, "Quantization loss in convolutional decoding," *IEEE Trans. Commun.*, vol. COM-41, pp. 261–265, Feb. 1993.

[74] W.-H.J. Chen, M.P.C. Fossorier, and S. Lin, "Quantization issues for soft-decision decoding of linear block codes," *IEEE Trans. Commun.*, vol. COM-47, pp. 789–795, June 1999.

[75] J.M. Wozencraft and I.M. Jacobs, *Principles of Communication Engineering*. New York: John Wiley & Sons, 1965.

[76] J.L. Massey, "Coding and modulation in digital communications," in *Proc. Int. Zurich Seminar on Digital Communications*, Mar. 1974, pp. E2(1)–E2(4).

[77] G. Taricco, "On the capacity of the binary input Gaussian and Rayleigh fading channels," *European Trans. Telecommun.*, vol. 7, pp. 201–208, Mar./Apr. 1996.

[78] N. Phamdo and F. Alajaji, "Soft-decision demodulation design for COVQ over white, colored, and ISI Gaussian channels," *IEEE Trans. Commun.*, vol. COM-48, pp. 1499–1506, Sept. 2000.

[79] X. Ma, X. Zhang, H. Yu, and A. Kavcic, "Optimal quantization for soft-decision decoding revisited," in *Proc. ISITA '02*, Oct. 2002, pp. 483–486.

[80] N. Binshtok and S. Shamai, "Integer metrics for binary input symmetric output memoryless channels," *IEEE Trans. Commun.*, vol. COM-47, pp. 1636–1645, Nov. 1999.

[81] M.R.G. Butler and A.R. Nix, "Quantization loss for convolutional decoding in Rayleigh-fading channels," *IEEE Commun. Letters*, vol. 7, pp. 446–448, Sept. 2003.

[82] K. Sayood, *Introduction to Data Compression.* San Francisco: Morgan Kaufmann Publishers, 1996.

[83] R.M. Gray and D.L. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. IT-44, pp. 2325–2383, Oct. 1998.

[84] L.-N. Lee, "On optimal soft-decision demodulation," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 437–444, July 1976.

[85] R.D. Parsons, Jr. and S.G. Wilson, "Polar quantizing for coded PSK transmission," *IEEE Trans. Commun.*, vol. COM-38, pp. 1511–1519, Sept. 1990.

[86] M.D. Ross and W.P. Osborne, "On the calculation of the exponential bound parameter for phase quantized 8-PSK," *IEEE Trans. Commun.*, vol. COM-43, pp. 674–676, Feb./Mar./Apr. 1995.

[87] H.J. Landau, "Sampling, data transmission, and the Nyquist rate," *IEEE Proceedings*, vol. 55, pp. 1701–1706, Oct. 1967.

[88] B.R. Saltzberg, "Intersymbol interference error bounds with application to ideal bandlimited signaling," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 563–568, July 1968.

[89] A. Fihel and H. Sari, "Performance of reduced-bandwidth 16 QAM with decision-feedback equalization," *IEEE Trans. Commun.*, vol. COM-35, pp. 715–723, July 1987.

[90] J. Salz, "Optimum mean-square decision feedback equalization," *Bell Syst. Tech. J.*, vol. 52, pp. 1341–1373, Oct. 1973.

[91] G.J. Foschini, "Contrasting performance of faster binary signaling with QAM," *Bell Syst. Tech. J.*, vol. 63, pp. 1419–1445, Oct. 1984.

[92] G.D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363–378, May 1972.

[93] G.D. Forney, "Lower bounds on error probability in the presence of large intersymbol interference," *IEEE Trans. Commun.*, vol. COM-20, pp. 76–77, Feb. 1972.

[94] J.E. Mazo, "Faster-than-Nyquist signaling," *Bell Syst. Tech. J.*, vol. 54, pp. 1451–1462, Oct. 1975.

[95] D. Hajela, "On computing the minimum distance for faster than Nyquist signaling," *IEEE Trans. Inform. Theory*, vol. IT-36, pp. 289–295, Mar. 1990.

[96] J.E. Mazo and H.J. Landau, "On the minimum distance problem for faster-than-Nyquist signaling," *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 1420–1427, Nov. 1988.

[97] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *European Trans. Telecommun.*, vol. 6, pp. 507–511, Sept./Oct. 1995.

[98] R. Karabed, P.H. Siegel, and E. Soljanin, "Constrained coding for binary channels with high intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-45, pp. 1777–1797, Sept. 1999.

[99] B.H. Marcus, P.H. Siegel, and J.K. Wolf, "Finite-state modulation codes for data storage," *IEEE J. Select. Areas Commun.*, vol. JSAC-10, pp. 5–37, Jan. 1992.

[100] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft decision outputs and its applications," in *Proc. GLOBECOM '89*, Nov. 1989, vol. 3, pp. 1680–1686.

[101] M.P.C. Fossorier, F. Burkert, S. Lin, and J. Hagenauer, "On the equivalence between SOVA and Max-Log-MAP decodings," *IEEE Commun. Letters*, vol. 2, pp. 137–139, May 1998.

[102] L. Ping, S. Chan, and K.L. Yeung, "Iterative decoding of multi-dimensional concatenated single parity check codes," in *Proc. ICC '98*, June 1998, vol. 1, pp. 131–135.

[103] J.G. Proakis, *Digital Communications.* New York: McGraw-Hill, 3rd edition, 1995.

[104] P. Kabal and S. Pasupathy, "Partial response signaling," *IEEE Trans. Commun.*, vol. COM-23, pp. 921–934, Sept. 1975.

[105] K.T. Wu and K. Feher, "Multi-level PRS/QPRS above the Nyquist rate," *IEEE Trans. Commun.*, vol. COM-33, pp. 735–739, July 1985.

[106] C. Wang and L. Lee, "Practically realizable digital transmission significantly

below the Nyquist bandwidth," *IEEE Trans. Commun.*, vol. COM-43, pp. 166–169, Feb./Mar./Apr. 1995.

[107] A. Said and J.B. Anderson, "Bandwidth-efficient coded modulation with optimized linear partial-response signals," *IEEE Trans. Inform. Theory*, vol. IT-44, pp. 701–713, Mar. 1998.

[108] F. Rusek and J.B. Anderson, "M-ary coded modulation by Butterworth filtering," in *Proc. ISIT '03*, Jun./Jul. 2003, p. 184.

[109] T.F. Wong, D.J. Skellern and L.H.C. Lee, "An automatic procedure to construct the optimal soft-decision quantizers and branch metrics for Viterbi decoders," in *Proc. URSI Int. Symp. Signals, Systems, and Electronics*, Oct. 1995, pp. 423–426.

APPENDIX A

DERIVATIVE OF MSE FOR UNIFORM SCALAR QUANTIZATION

Uniformly Spaced Levels

In this appendix we will show how setting the derivative of the MSE $D^u_{B,L}$ with respect to the uniform scalar quantizer step τ yields equations (3.14) and (3.15) assuming uniform spacing between the quantizer levels.

Since for uniform scalar quantization $a_j = \left(j - \frac{L}{2}\right)\tau$, for $i = 0$ (3.6) can be rewritten as (3.15).

Taking the derivative of (3.5) yields

$$\frac{\partial D^u_{B,L}}{\partial \tau} = -2\sum_{j=0}^{L-1} v'_j\, c_{0j} - 2\sum_{j=0}^{L-1} v_j\, c'_{0j} + 2\sum_{j=0}^{L-1} v_j\, v'_j\, P_{0j} + \sum_{j=0}^{L-1} v_j^2\, P'_{0j} \tag{A.1}$$

where $v'_j = \frac{\partial v_j}{\partial \tau}$, $c'_{0j} = \frac{\partial c_{0j}}{\partial \tau}$, and $P'_{0j} = \frac{\partial P_{0j}}{\partial \tau}$.

Now, from equation (3.2), P_{0j} can be written as follows, for a uniform scalar quantizer with $a_j = \left(j - \frac{L}{2}\right)\tau$ for $j = 1, 2, \ldots, L-2$,

$$\begin{aligned} P_{0j} &= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{\left(j-\frac{L}{2}\right)\tau}^{\left(j+1-\frac{L}{2}\right)\tau} \mathrm{e}^{-\frac{(u-1)^2}{2\sigma^2}}\, du \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\left(j+1-\frac{L}{2}\right)\tau} \mathrm{e}^{-\frac{(u-1)^2}{2\sigma^2}}\, du - \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\left(j-\frac{L}{2}\right)\tau} \mathrm{e}^{-\frac{(u-1)^2}{2\sigma^2}}\, du \end{aligned} \tag{A.2}$$

while P_{00} is

$$P_{00} = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\left(1-\frac{L}{2}\right)\tau} \mathrm{e}^{-\frac{(u-1)^2}{2\sigma^2}}\, du \tag{A.3}$$

and $P_{0(L-1)}$ is

$$P_{0(L-1)} = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{\left(\frac{L}{2}-1\right)\tau}^{\infty} \mathrm{e}^{-\frac{(u-1)^2}{2\sigma^2}}\, du = 1 - \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\left(\frac{L}{2}-1\right)\tau} \mathrm{e}^{-\frac{(u-1)^2}{2\sigma^2}}\, du \tag{A.4}$$

So, P'_{0j} can be written as

$$P'_{0j} \;=\; \frac{j+1-\frac{L}{2}}{\sqrt{2\pi\sigma^2}}\, \mathrm{e}^{-\frac{\left[\left(j+1-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} - \frac{j-\frac{L}{2}}{\sqrt{2\pi\sigma^2}}\, \mathrm{e}^{-\frac{\left[\left(j-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} \tag{A.5}$$

for $j = 1, 2, \ldots, L-2$, while P'_{00} is

$$P'_{00} = \frac{1-\frac{L}{2}}{\sqrt{2\pi\sigma^2}}\, \mathrm{e}^{-\frac{\left[\left(1-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} \tag{A.6}$$

and $P'_{0(L-1)}$ is

$$P'_{0(L-1)} = -\frac{\frac{L}{2}-1}{\sqrt{2\pi\sigma^2}}\, \mathrm{e}^{-\frac{\left[\left(\frac{L}{2}-1\right)\tau-1\right]^2}{2\sigma^2}} \tag{A.7}$$

Using the last three equations and regrouping inside the sum $\sum_{j=0}^{L-1} v_j^2\, P'_{0j}$, we get the equivalent expression

$$\begin{aligned} \sum_{j=0}^{L-1} v_j^2\, P'_{0j} \;&=\; \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{j=1}^{L-1} \left(v_{j-1}^2 - v_j^2\right) \left(j - \frac{L}{2}\right) \mathrm{e}^{-\frac{\left[\left(j-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} \\ &=\; -\,\frac{2\,\tau^2}{\sqrt{2\pi\sigma^2}} \sum_{j=1}^{L-1} \left(j - \frac{L}{2}\right)^2 \mathrm{e}^{-\frac{\left[\left(j-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} \end{aligned} \tag{A.8}$$

where in the last step we used the fact that $v_j = \left(j - \frac{L-1}{2}\right)\,\tau$.

Working similarly with c'_{0j} we can get

$$c'_{0j} \;=\; \frac{\left(j+1-\frac{L}{2}\right)^2\,\tau}{\sqrt{2\pi\sigma^2}}\, \mathrm{e}^{-\frac{\left[\left(j+1-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} - \frac{\left(j-\frac{L}{2}\right)^2\,\tau}{\sqrt{2\pi\sigma^2}}\, \mathrm{e}^{-\frac{\left[\left(j-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} \tag{A.9}$$

for $j = 1, 2, \ldots, L-2$, while c'_{00} is

$$c'_{00} = \frac{\left(1-\frac{L}{2}\right)^2\,\tau}{\sqrt{2\pi\sigma^2}}\, \mathrm{e}^{-\frac{\left[\left(1-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} \tag{A.10}$$

and $c'_{0(L-1)}$ is

$$c'_{0(L-1)} = -\frac{\left(\frac{L}{2}-1\right)^2 \tau}{\sqrt{2\pi\sigma^2}}\, \mathrm{e}^{-\frac{\left[\left(\frac{L}{2}-1\right)\tau-1\right]^2}{2\sigma^2}} \tag{A.11}$$

Using the last three equations and regrouping inside the sum $\sum_{j=0}^{L-1} v_j\, c'_{0j}$, we get the equivalent expression

$$\begin{aligned} \sum_{j=0}^{L-1} v_j\, c'_{0j} &= \frac{\tau}{\sqrt{2\pi\sigma^2}} \sum_{j=1}^{L-1} (v_{j-1} - v_j) \left(j - \frac{L}{2}\right)^2 \mathrm{e}^{-\frac{\left[\left(j-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} \\ &= -\frac{\tau^2}{\sqrt{2\pi\sigma^2}} \sum_{j=1}^{L-1} \left(j - \frac{L}{2}\right)^2 \mathrm{e}^{-\frac{\left[\left(j-\frac{L}{2}\right)\tau-1\right]^2}{2\sigma^2}} \end{aligned} \tag{A.12}$$

From this last equation and (A.8) we get that

$$\sum_{j=0}^{L-1} v_j^2\, P'_{0j} = 2 \sum_{j=0}^{L-1} v_j\, c'_{0j} \tag{A.13}$$

so these two sums are canceled in (A.1) yielding

$$\begin{aligned} \frac{\partial D^u_{B,L}}{\partial \tau} &= -2 \sum_{j=0}^{L-1} v'_j\, c_{0j} + 2 \sum_{j=0}^{L-1} v_j\, v'_j\, P_{0j} \\ &= 2 \sum_{j=0}^{L-1} \left(j - \frac{L-1}{2}\right) \left[\left(j - \frac{L-1}{2}\right) \tau\, P_{0j} - c_{0j}\right] \end{aligned} \tag{A.14}$$

where we used the fact that $v_j = \left(j - \frac{L-1}{2}\right)\tau$ and thus, $v'_j = j - \frac{L-1}{2}$. Setting this last expression of $\frac{\partial D^u_{B,L}}{\partial \tau}$ equal to zero, we get equation (3.14).

Centroid Levels

In this part of this Appendix we will show how taking the derivative of the uniform MSE $D^u_{B,L}$ with respect to the uniform scalar quantizer step τ yields equation (3.16), assuming centroid quantization levels. It is straightforward to find that the centroid levels v_j can be expressed as

$$v_j = \frac{c_{0j} + c_{1j}}{P_{0j} + P_{1j}} = \frac{c_{0j} - c_{0(L-1-j)}}{P_{0j} + P_{0(L-1-j)}} \tag{A.15}$$

where the P_{ij} and the c_{ij} are defined in equations (3.2) and (3.6), respectively, and in the last step we used the fact that due to the quantizer symmetry $P_{1j} = P_{0(L-1-j)}$ and $c_{1j} = -c_{0(L-1-j)}$.

Because of this last expression for the quantization levels v_j, we can rewrite the general MSE $D_{B,L}$ of equation (3.4) as

$$D_{B,L} \quad = \quad 1 + \sigma^2 - \frac{1}{2} \sum_{j=0}^{L-1} \frac{\left(c_{0j} + c_{1j}\right)^2}{P_{0j} + P_{1j}} \tag{A.16}$$

So far we have not made any assumptions for uniform quantization and therefore, we can use this last expression in more cases as in the case of nonuniform MSE-minimizing quantization.

Now, assuming uniformly spaced quantization boundaries $a_j = \left(j - \frac{L}{2}\right) \tau$ for $j = 1, 2, \ldots, L-1$ and taking the derivative of the last expression of the MSE $D^u_{B,L}$ with respect to τ yields

$$\begin{aligned} \frac{\partial D^u_{B,L}}{\partial \tau} &= \sum_{j=0}^{L-1} v_j \left(c'_{0j} + c'_{1j}\right) - \frac{1}{2} \sum_{j=0}^{L-1} v_j^2 \left(P'_{0j} + P'_{1j}\right) \\ &= 2 \sum_{j=0}^{L-1} v_j \, c'_{0j} - \sum_{j=0}^{L-1} v_j^2 \, P'_{0j} \end{aligned} \tag{A.17}$$

where in the last step we used the quantizer's symmetry around zero.

Using the expressions of P'_{0j} and c'_{0j} from Appendix A and rearranging inside the sums of the last expression for the $\frac{\partial D^u_{B,L}}{\partial \tau}$ we get

$$\begin{aligned}
\frac{\partial D^u_{B,L}}{\partial \tau} &= 2 \sum_{j=1}^{L-1} (v_{j-1} - v_j) \left(j - \frac{L}{2}\right)^2 \tau \, e^{-\frac{\left[\left(j-\frac{L}{2}\right)\tau - 1\right]^2}{2\sigma^2}} \\
&\qquad - \sum_{j=1}^{L-1} \left(v_{j-1}^2 - v_j^2\right) \left(j - \frac{L}{2}\right) e^{-\frac{\left[\left(j-\frac{L}{2}\right)\tau - 1\right]^2}{2\sigma^2}} \\
&= \sum_{j=1}^{L-1} (v_{j-1} - v_j) \left(j - \frac{L}{2}\right) \left[2\left(j - \frac{L}{2}\right)\tau - v_{j-1} - v_j\right] e^{-\frac{\left[\left(j-\frac{L}{2}\right)\tau - 1\right]^2}{2\sigma^2}}
\end{aligned} \tag{A.18}$$

Setting the last expression equal to zero and dropping some common terms results in (3.16).

APPENDIX B

DERIVATIVE OF CUTOFF RATE FOR UNIFORM SCALAR QUANTIZATION

In this appendix we will show how setting the derivative of the cutoff rate $R_{B,L}$ with respect to the uniform scalar quantizer step τ yields equation (3.17).

Starting from equation (3.9), we get

$$\frac{\partial R_{B,L}}{\partial \tau} = -\frac{1}{\log 2} \frac{\sum_{j=0}^{L-1} \frac{P'_{0j}P_{1j}+P_{0j}P'_{1j}}{2\sqrt{P_{0j}P_{1j}}}}{1+\sum_{j=0}^{L-1}\sqrt{P_{0j}P_{1j}}} \tag{B.1}$$

where $P'_{ij} = \frac{\partial P_{ij}}{\partial \tau}$. Setting the derivative $\frac{\partial R_{B,L}}{\partial \tau}$ equal to zero, we can simplify the above expression to get the equation

$$\sum_{j=0}^{L-1} \frac{P'_{0j}P_{1j}+P_{0j}P'_{1j}}{\sqrt{P_{0j}P_{1j}}} = 0 \tag{B.2}$$

Defining $\kappa_j = \sqrt{\frac{P_{1j}}{P_{0j}}}$, we can rewrite the last equation as

$$\sum_{j=0}^{L-1} \left(P'_{0j}\,\kappa_j + P'_{1j}\,\frac{1}{\kappa_j} \right) = 0 \tag{B.3}$$

Due to the quantizer symmetry, $P_{0j} = P_{1(L-1-j)}$ and thus, $P'_{0j} = P'_{1(L-1-j)}$ and $\kappa_j = \frac{1}{\kappa_{L-1-j}}$. Therefore, $\sum_{j=0}^{L-1} P'_{1j}\,\frac{1}{\kappa_j} = \sum_{j=0}^{L-1} P'_{0j}\,\kappa_j$ and equation (B.3) is further simplified to

$$\sum_{j=0}^{L-1} P'_{0j}\,\kappa_j = 0 \tag{B.4}$$

Substituting the three expressions (A.2)-(A.4) of P_{0j} in the sum on the left side of (B.4) and regrouping before differentiating we get

$$\begin{aligned}
\sum_{j=0}^{L-1} P'_{0j}\,\kappa_j &= \sum_{j=1}^{L-1} (\kappa_{j-1} - \kappa_j)\, \frac{\partial}{\partial \tau} \left(\frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\left(j-\frac{L}{2}\right)\tau} \mathrm{e}^{-\frac{(u-1)^2}{2\sigma^2}}\, du \right) \\
&= \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{j=1}^{L-1} (\kappa_{j-1} - \kappa_j) \left(j - \frac{L}{2} \right) \mathrm{e}^{-\frac{\left[\left(j-\frac{L}{2}\right)\tau - 1\right]^2}{2\sigma^2}} \qquad \text{(B.5)}
\end{aligned}$$

By setting the last expression equal to zero, and canceling whatever can be factored out of the sum, we get (3.17).

APPENDIX C

VARIABLE-RATE SCALAR QUANTIZATION AND VECTOR QUANTIZATION

Variable-Rate Scalar Quantization

The scalar quantizers used in Chapter III are called fixed-rate because the quantizer output rate is measured as the logarithm of the number of the quantization levels, i.e., $q = \log_2(L)$. Variable-rate scalar quantization, also called entropy-coded scalar quantization, is a simple way of improving the performance of a fixed-rate scalar quantizer [83]. It refers to a scalar quantizer followed by a variable rate encoder, e.g., a Huffman encoder, that takes into account the nonequal probabilities of each quantizer level to obtain significant rate savings. The entropy of the quantizer output levels is the lower bound on the rate that an variable rate code can achieve, and since a Huffman code can be shown to come quite close to this limit [83], the quantizer output rate is given by the entropy of the quantizer output.

It is not difficult to realize that variable-rate scalar quantization can yield some throughput gains for higher SNRs, since it allows a larger number of levels in the scalar quantizer. Limiting both the fixed-rate and the variable-rate scalar uniform quantizer to at most 2 bits the results are shown in Fig. 66. For both quantizers the mutual information maximization criterion has been used.

The design process for such quantizers is more involved, as it has to include a Lagrange multiplier, and the addition of the variable-rate code makes the quantizer more complex. Furthermore, to exploit the gains of Fig. 66, the number of levels in the quantizer has to be changed for different SNR, which makes variable-rate quantization nonpractical and excludes any notion of robustness.

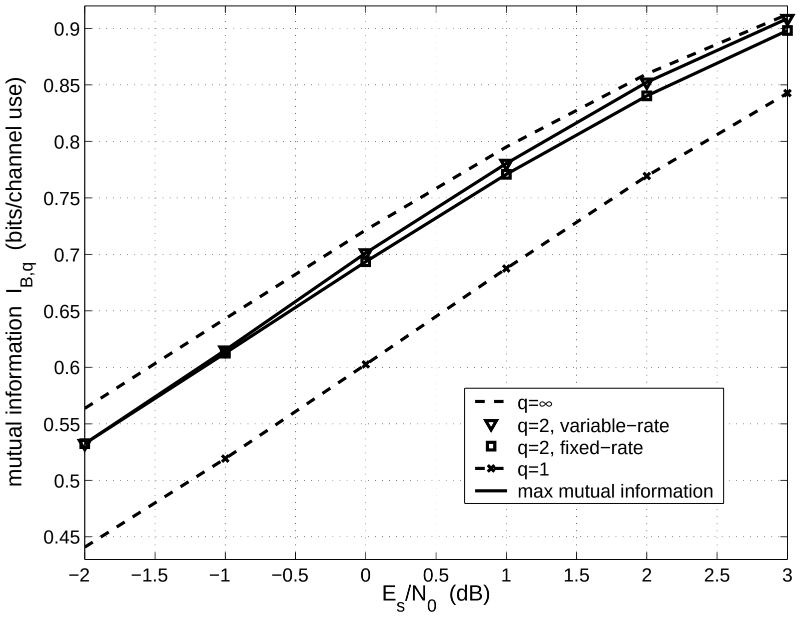

Fig. 66. Maximum mutual information $I_{B,q}$ as a function of E_s/N_0 for a fixed-rate uniform scalar quantizer with $L = 2^q = 2, 4$ levels and for a variable-rate uniform scalar quantizer with rate $q = 2$ bits.

Vector Quantization

Although variable-rate uniform scalar quantization provided us with some gains, it could not improve the fixed-rate uniform scalar quantization throughput for low channel SNRs. At these low SNRs the output of the BIAWGN channel, i.e., the bimodal Gaussian pdf, looks more uniform and thus, the statistics of the quantization levels cannot yield rate savings.

However, in conventional quantization there are ways to improve the quantizer performance even for uniform sources by quantizing the source samples in groups of k instead of one at a time [83]. This type of performance gain is associated with the distortion metric used to measure the quantizer's performance in representing a source and not with the statistics of the source [83]. Such quantizers are called vector quantizers and both fixed-rate and variable-rate vector quantization has been considered and analyzed before.

Here we are not dealing with a conventional quantization problem as the distortion metric, i.e., the mutual information or the cutoff rate of the equivalent channel, is associated only with the probabilities of each quantization region and not with the values of the quantizer levels. Hence, whether such a "space-filling" gain from vector quantization exists, does not appear to have an intuitive answer.

Considering just two-dimensional quantization, i.e., two channel measurements are quantized at the same time, the quantizer designs should fill the two-dimensional space at least as good as uniform scalar quantizers. One should note that the equivalent channel of a system with a 2-D quantizer is different from the one shown in Fig. 31. Now the equivalent channel has two input bits, instead of just one, and the equivalent channel output, i.e., the 2-D quantizer output, has $L^2 = 2^{2q}$ levels, so that the quantizer output rate equals $\frac{2q}{2} = q$ bits per quantized channel measurement.

This problem seems to be equivalent to designing a limited precision demodulator for a 4-QAM or QPSK constellation with the objective to maximize the mutual information rate. This problem has been studied before [84, 85, 86, 109], in the cutoff rate maximizing context.

Lee [84] established a necessary condition for the optimality in the cutoff-rate sense for the quantizer boundaries. According to this condition, for all points $\boldsymbol{r}$ on the boundary between two adjacent quantization regions A and B the following equality must hold:

$$\sum_{m=0}^{M-1} \left[\frac{1}{\sqrt{P(B|\boldsymbol{x}^m)}} \sum_{i=0}^{M-1} \sqrt{P(B|\boldsymbol{x}^i)} - \frac{1}{\sqrt{P(A|\boldsymbol{x}^m)}} \sum_{i=0}^{M-1} \sqrt{P(A|\boldsymbol{x}^i)} \right] p(\boldsymbol{r}|\boldsymbol{x}^m) = 0 \tag{C.1}$$

with $M = 2^n$ being the number of the different messages that can be transmitted as

n bits are considered together, i.e., n is the dimension of the vector quantizer, and $\boldsymbol{x}^m$ is the binary length-n representation of the message m. In other words, we only consider Lee's necessary condition [84] for constellations of 1 bit/dimension (simplex signals).

When using a scalar quantizer it can be shown that if the scalar quantizer satisfies Lee's necessary condition for an one-dimensional constellation, then any n-dimensional vector quantizer that is the cartesian product of this scalar quantizer also satisfies this necessary condition for the n-dimensional constellation that is the cartesian product of the one-dimensional constellation. This means that the scalar quantizer results in a local maximum for the cutoff rate even if we allow for higher quantizer dimensionality. What is not possible to show is whether this local minimum is the global one or not. We tried several two-dimensional quantizers for the BIAWGN channel but could never exceed the throughput of a scalar quantizer.

The practical problems with vector quantization is that it correlates the two or more channel measurements that are quantized together and this should be taken into account in the channel code design. In addition, the channel decoding algorithm should take this correlation into account and thus, becomes more complex.

VITA

Angelos Dimitriou Liveris received the Bachelor of Science degree in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 1998. Since 1999, he has been working toward the Ph.D. degree in electrical engineering at Texas A&M University, College Station, TX. His research interests include network information theory, distributed source and channel coding, detection and wireless communications. Angelos Liveris can be contacted at Stefanou Vyzantiou 44, Athens 11144, Greece.