JOINT AND TANDEM SOURCE-CHANNEL CODING WITH COMPLEXITY AND DELAY CONSTRAINTS

by

Jongtae Lim

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2001

Doctoral Committee:

Professor David L. Neuhoff, Chairman
Assistant Professor Mingyan Liu
Professor Kang G. Shin
Professor Wayne E. Stark

To Hyunseon,

Jenny Yewon & Daniel Yejun

ACKNOWLEDGMENTS

Most of all, I give thanks to the Lord with all my heart, for He is always with me and He always loves me. He blessed me to meet my advisor, Professor David Neuhoff, for whom I have my deepest respect and gratitude. Without his kindhearted guidance, encouragement and patience with me, my work would not have been possible. Also, my thanks go to Professor Wayne E. Stark, who gave me valuable suggestions and encouragement throughout the course of my studies. My gratitude also goes to Professors Kang Geun Shin and Mingyan Liu for reviewing this thesis in spite of their busy schedules. Also, I give special thanks to Troy C. Nolan who offered a valuable suggestion.

I have been financially supported by the Department of Defense Research & Engineering (DDR&E) Multidisciplinary University Research Initiative (MURI) on Low Energy Electronics Design for Mobile Platforms under grant ARO DAAH04-96-1037 which has been managed by the Army Research Office (ARO).

I want to express my great thanks and love to my lovely wife, Hyunseon, who has stood by me all the way and has comforted me in all areas of my life. Finally, I give my love to my daughter, Jenny, and my son, Daniel, who make me feel the joy of life.

TABLE TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

CHAPTER I

INTRODUCTION

In traditional communication systems where an information source is transmitted over a noisy channel through cascaded source and channel coding, source and channel codes have usually been designed independently. Source codes have been designed to have good performance under noiseless channel conditions, i.e. without consideration of channel noise, and channel codes have been designed to reduce the effect of channel noise without regard to the nature of the information source and the source code. The achievability of optimal performance using the tandem (cascaded) connection of separately designed source and channel codes is asserted by Shannon's source and channel separation theorem [1]. However, the theorem is an asymptotic result involving arbitrarily large source code dimension and infinite channel code block length, and hence unlimited complexity and delay. For practical systems having limitations on complexity and delay, this separation theorem may not be applicable. Thus, there has been considerable interest in joint source-channel coding, where source and channel code are jointly designed to combat channel errors, with the motivation that it can achieve better performance with less complexity and/or delay than separately designed tandem source-channel coding.

Since Kurtenbach and Wintz [2] introduced a method of designing scalar quantizers over noisy channels, many designers of source coders have considered methods for designing robust quantizers over noisy channels without any explicit channel codes. These approaches modify quantizers taking into account the characteristic of the noisy channel and minimizing distortion caused by quantization and channel errors. In these approaches the distortion measure is replaced by the expected distortion over the noisy channel. Quantization using these methods is called “channel-optimized quantization”. Kumazawa, Kasahara and Namekawa [3] introduced a Lloyd algorithm for vector quantizers using the modified distortion measure. An iterative design algorithm for a channel-optimized scalar quantizer over noisy channels is provided by Farvardin and Vaishampayan [4]. Their algorithm is extended to block transform coding in [5], and they reinvented channel-optimized vector quantization [6]. They showed that optimizing quantizers for noisy channels improves the performance without any added complexity. It is noticeable that as the channel becomes noisier, the total number of quantization cells becomes smaller than the total number of available codewords. These decrements mean a reduction in source code rate and can be considered as implicit channel coding.

Other designers of source coders have focused on optimizing the index assignment: mapping the quantized source symbols to channel code symbols in such a way that the overall mean squared error is minimized over all possible index assignments. This approach focuses on minimizing the distortion caused by likely channel codeword errors by coding the indices of quantized source codewords in such a way that channel codewords having small Hamming distance are associated with reproduction codewords with small Euclidean distance. Rydbeck and Sundberg [7] studied several index assignment methods for scalar quantizers, and DeMarca and Jayant [8] introduced an iterative search algorithm for good index assignments for scalar quantizers. Zeger and Gersho [9] extended the iterative search algorithm to vector quantization. For the uniform source and uniform quantizer on a binary symmetric channel, the optimality of the natural binary assignment was shown in [10]. Later, this result was rederived and generalized by

McLanghin and Neuhoff [11]. Recently, Mehes and Zeger [12] considered optimal index assignment for a binary asymmetric channel. They showed that the natural binary assignment is not optimal on the binary asymmetric channel for uniform scalar quantization.

While channel-optimized quantization and optimal index assignment are approaches to joint source-channel coding in the context of channel-optimized source coding, a variety of approaches fall into the category of source-optimized channel coding. This approach is to design the source code without regard to channel noise, and then design the channel code for the designed source code and a given channel. A common example of this approach is unequal error protection, which protects the source code with different protection levels according to the characteristics of the source code. For example, in transform coding, transform coefficients are encoded using several channel codes with different protection levels for the coefficients with different levels of importance. In some cases, the bits of each codeword index are unequally protected to get better performance. For example, in uniform scalar quantization, the distortion can be reduced by assigning a channel code with a lower rate to the most significant bit of the codeword, because a bit error in the most significant bit causes more distortion than in the least significant bit. The method of unequal error protection has many variations according to the channel code and the channel. Modestino and Daut [13] considered this problem in differential pulse code modulation (DPCM) image coding. By providing error control protection to those encoded bits that contribute most significantly to image reconstruction, it was possible to minimize the degradation caused by channel errors without sacrificing transmission bandwidth. They extended this method to block transform image coding in [14]. Since Hagenauer [15] introduced an application of rate-compatible punctured convolutional codes (RCPC) to unequal error protection, many researchers have used RCPC codes to provide unequal error protection. RCPC codes allow convolutional codes of several different rates to be used with only one encoder and one decoder, and to easily change its rate by varying the number of punctured bits in the puncturing matrix. Tanabe and Farvardin [16] applied this idea to subband image coding. Since the convolutional codes can correct long burst errors as well as a few random errors, RCPC codes are good for use in providing protection against channel noise for channels with burst errors such as wireless channels. For example, an RCPC channel code matched to subband speech coding is designed for a Rayleigh fading channel [17]. Recent work is found in [18] and [19], where a progressive Said-Pearlman image coder is protected by an RCPC channel code.

An approach combining channel-optimized source coding and source-optimized channel coding is considered by Goldsmith and Effros [20]. Their joint designs use channel-optimized vector quantization (COVQ) for the source code and RCPC coding for the channel code. They use an iterative design technique to minimize the overall distortion. Specifically, for a given source code rate, the COVQ is optimized for the existing RCPC code and then the RCPC code is optimized for the optimized COVQ. This optimization is continued until it converges to a local minimum and this iterative design process is repeated for the possible values of source code rate less than the channel transmission rate, which is defined as the number of channel uses per source sample. They compared the performance of this technique with those of a source-optimized channel code using a vector quantizer and an RCPC channel code, and a channel-optimized source code using COVQ and an equal-error-protection convolutional channel code. They found that all three code designs have roughly the same performance when their bit allocations are optimized. Thus they concluded that the most important aspect for any of these source and channel code designs is the optimal bit allocation between the source and channel coders.

Other approaches to joint source-channel coding include iterative designs using trellis-coded quantization (TCQ) [21], [22], [23], joint design of source coder and modulation signal set [24], unequal error protection using multicarrier modulation, joint optimizing of channel-optimized quantizer and multicarrier modulation [25], and so on.

A fundamental problem in joint source-channel coding is the choice of source code rate (or channel code rate) that optimally allocates the available transmission rate between source and channel coding. As the source code rate increases, the distortion due to lossy source coding (i.e. quantization noise) is decreased, but the redundancy in the channel code decreases, and thus the distortion caused by channel noise increases. This tradeoff between source and channel coding is analyzed theoretically by Hochwald and Zeger [26]. They established asymptotic upper and lower bounds on the channel code rate that minimizes the average distortion of a vector quantizer cascaded with a channel coder and a binary symmetric channel using high-resolution quantization theory. Hochwald expanded their analysis to Gaussian channels [27].

Although many studies in joint source-channel coding have the motivation that the joint source-channel coding can achieve better performance with less complexity and delay than tandem source-channel coding, little quantitative evidence for this claim has appeared in the literature. In this thesis, we search for such evidence by quantitatively comparing representative systems of each type on the basis of distortion vs. complexity and distortion vs. delay.

To avoid idiosyncrasies, we examine systems that are as "plain vanilla" as possible. Specifically, as the joint source-channel code, we consider the channel-optimized transform code studied by Vaishampayan and Favardin [5]. As the tandem source-channel code, we consider a conventional transform code followed by a Reed-

Solomon channel code. We do not use entropy coding, because it is very sensitive to channel errors, and although there are ways to mitigate such sensitivity, the performance of such methods tends to be highly method specific, making it unlikely that we could choose a representative plan vanilla version. We also choose mean-squared error (MSE) as the fidelity measure which is by far the most commonly used. For each system, formulas for the mean-squared error, complexity and delay are found and used to minimize distortion subject to a complexity or delay constraint. The complexity of these systems is estimated by counting the numbers of arithmetic operations per data sample required for encoding and decoding, and the delay is expressed in units of data samples by considering buffering delay caused by block processing. We evaluate the performance of these systems on a Gauss-Markov (first-order AR) source over binary symmetric channels (BSC).

One of the principal contributions of this thesis is our analysis of the performance of joint and tandem source-channel coding. Our results suggest there exists a complexity threshold such that if the number of available operations per sample is above the threshold, then tandem source-channel coding is better, while if the number of available operations per sample is below the threshold, then joint source-channel coding is better.

Like the results with complexity constraints, our results with delay constraints suggest there is a delay threshold such that tandem source-channel coding is better than joint source-channel coding when and only when the permissible delay is above the threshold.

Our investigation is also expanded to unequal error protection in tandem source-channel coding system. The transform coefficients of transform coders have different levels of importance. That is, the distortion caused by a bit error in one coefficient is somewhat different from that caused by a bit error in other coefficients. Thus if we provide different levels of error protection by varying the amount of channel coding according to the importance of the transform coefficient, improved performance can be obtained. We apply this unequal error protection scheme to the tandem system and investigate the performance of unequal error protection system with a complexity constraint.

The results show that the unequal error protection scheme improves the performance of tandem source-channel coding system under a given complexity constraint and thus it reduces the complexity threshold relative to that of an equal error protection system.

A fundamental problem in designing transform codes is the rate allocation problem, in which the given rate should be allocated among the coefficients to minimize the overall distortion. Design algorithms for optimal rate allocation are well understood for noiseless channels, and various methods and analytical formulas are found in [32]. However, the rate allocation problem for noisy channel has not been well studied. In this thesis, based on high-resolution quantization theory we derive an analytic formula for the optimal bit allocation for noisy channels

The resulting asymptotically optimal rate allocation for noisy channels suggests that more bits need to be allocated to the coefficients whose variances are greater than the geometric mean of the coefficients variances, compared to the optimal bit allocation for noiseless channels. On the other hand, fewer bits are to be allocated to the coefficients whose variances are smaller than the geometric mean. The correspondence between experimental results and the asymptotic expression is well observed.

The rest of this thesis is organized as follows. Chapter 2 describes a generic communication system and the basic operation of the system. Introducing Shannon's rate-distortion function and channel capacity, the best possible performance of the system is presented. Also, transform coding is introduced and its theoretical performance limits are reviewed.

Chapter 3 is dedicated to the comparison of joint and tandem source-channel coding with complexity constraints and Chapter 4 expands the investigation to unequal error protection in tandem source-channel coding system. The comparison under delay constraints is given in Chapter 5.

A high-resolution analytic formula for the optimal rate allocation for noisy channels is given in Chapter 6. Finally, Chapter 7 provides concluding remarks and suggests some interesting future work.

CHAPTER II
COMMUNICATION SYSTEM AND THEORETICAL PERFORMANCE

The purpose of this chapter is to introduce tandem and joint source-channel coding systems, and to review Shannon's theory giving the theoretical limits to their performance. Transform coding examples of each type of system, and their performance limits, are also reviewed.

2.1 The Source-Channel Coding Systems

For communicating analog data samples across a noisy channel, we encounter two common approaches to source-channel coding: tandem source-channel coding and joint source-channel coding. This section describes the basic structure of these systems.

2.1.1 Tandem Source-Channel Coding System

A communication system in which a source code and a channel code are cascaded is called a tandem source-channel coding system. Figure 2.1 shows the basic elements of a tandem source-channel coding system: a source, a source encoder, a channel encoder, a modulator, a channel, a demodulator, a channel decoder and a source decoder. The source encoder is followed by the channel encoder.

The source is modeled as a stationary random sequence denoted $\{U_i\}$, which is

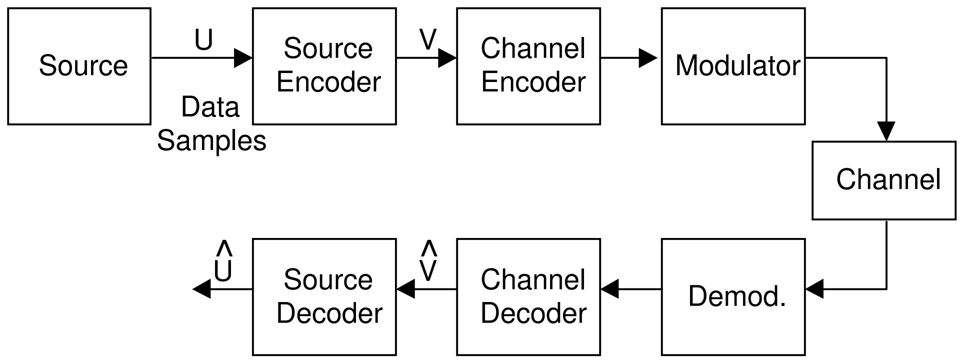


Figure 2.1: The basic elements of a generic communication system

input to the source encoder. The source encoder efficiently compresses the data samples from the source and converts it to a sequence V of symbols which is usually a sequence of binary digits. The length of the sequence V may not be the same for all source samples. When the lengths are the same for all source samples, the encoder is referred to as fixed-length or fixed-rate. Otherwise it is called a variable-length or variable-rate encoder. The channel encoder adds redundancy to this sequence to combat the channel noise. The output of the channel encoder, i.e. a channel codeword, is mapped to appropriate signal waveforms by the modulator and transmitted through the channel. These waveforms are corrupted by the channel noise presented in the channel. At the receiving end of the system, the demodulator processes the transmitted waveforms corrupted by the channel, determines which waveforms were transmitted, and puts out the corresponding sequence as an estimate of the transmitted channel codeword. The channel decoder removes the redundancy from this sequence and passes the output sequence $\hat{V}$ into the source decoder. The source decoder forms a reconstruction $\{\hat{U}_i\}$ of the original source samples

$\{U_i\}$. Due to channel noise, channel decoding errors, and distortion caused by the source encoder, the reconstructed source samples $\{\hat{U}_i\}$ are estimates of the original source samples. The fidelity of the reconstructed samples with respect to the original samples is a measure of the overall performance in the communication system. The mean-squared-error between the reconstructed samples and the original samples is commonly used as the fidelity measure.

Often, the above communication system is simplified as in Figure 2.2, where the modulator, the channel and the demodulator are modeled as a binary symmetric channel (BSC) whose input and output alphabets consist of binary elements (0 and 1). For example, a binary phase-shift keying (BPSK) modulation/demodulation and an additive white Gaussian noise (AWGN) channel are considered as a binary symmetric channel. A discrete channel is said to be symmetric if the probability that a transmitted symbol will be received incorrectly is independent of the value of the symbol, and a discrete channel is said to be memoryless if the probability distribution of the output depends only on the input at that time and is conditionally independent of previous channel inputs or outputs.

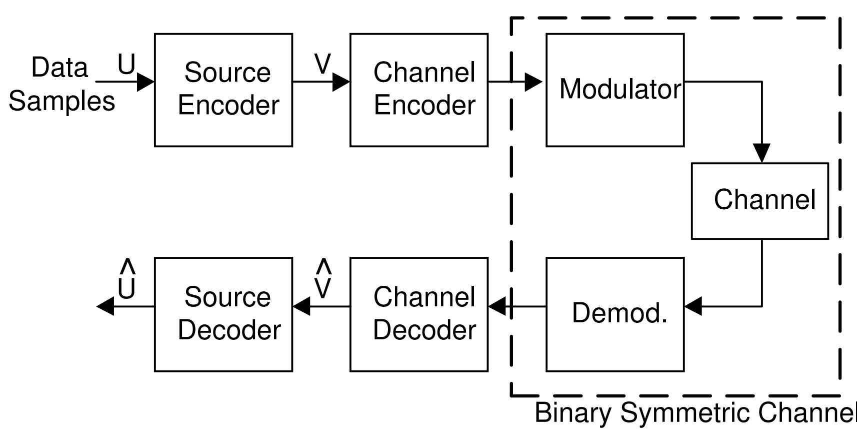


Figure 2.2: The simplified communication system with a Binary Symmetric Channel

The BSC is characterized by its crossover (or error) probability p, which is the conditional probability of receiving the other symbol given that one symbol was transmitted.

In a tandem source-channel coding system, we refer to several rates: the source code rate, channel code rate and transmission rate. The source code rate, R_S, is defined as the number of bits per source sample produced by the source coder. When a channel encoder transforms each k-bit data block into a larger block of n bits, the ratio of data bits to total bits, k/n, is called the channel code rate, R_C. The channel code rate can be thought of as the fraction of channel bits that constitutes information (i.e. bits from the source code). For example, in a channel code with rate ½, each coded bit carries ½ bit of information. The number of channel uses per data sample is called the transmission rate, R. For binary channels, we have the relationship of $R = R_S / R_C$.

2.1.2 Joint Source-Channel Coding System

In the tandem source-channel coding system, the source code and channel code are designed separately and operated in tandem. Since Shannon showed by his source-channel separation theorem [1] that no system could be better than the best tandem source-channel coding system, many communication systems have consisted of tandem connection of independently designed source and channel codes. However, the separation theorem cannot be applied to applications with complexity and delay constraints since it is based on unlimited source code dimension and channel code block length. This has motivated the combining of the source and channel codes to get better performance with less complexity and delay. In other words, it has established a new strategy, joint source-channel coding, in which the source and channel codes are combined and designed together.

Figure 2.3 illustrates the basic elements of the joint source-channel coding system consisting of an encoder, a channel and a decoder. The source, modeled as a stationary random sequence $\{U_i\}$, is converted to a sequence $\{V_i\}$ by the encoder which is intended to be robust to channel errors produced by the channel. The decoder produces reproductions $\{\hat{U}_i\}$ of the transmitted source samples.

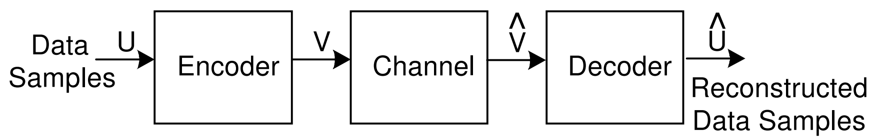


Figure 2.3: Basic Elements of Joint Source-Channel Coding System

2.2 Theoretical Performance Limits

The usual performance measure for the communication system in the previous section is the mean-squared error (MSE) between the source samples $\{U_i\}$ and their reproductions $\{\hat{U}_i\}$. Shannon's rate-distortion theory and channel coding theory determine the fundamental limits on performance of such systems. Shannon's distortion-rate function $D(R_S)$ gives the fundamental distortion limit when the source is encoded with source code rate less than or equal to R_S bits per sample. For example, for the first-order autoregressive Gaussian source, the distortion limit is given by the following distortion-rate function:

$$D(R_S) = \sigma^2(1-\rho^2)2^{-2R_S}, \text{ for } R_S \geq R_o, \tag{2.1}$$

where σ^2 is the variance of the source, ρ is correlation coefficient and $R_0 = \log_2(1+\rho)$

[27]. Along with the mean-squared error, signal-to-noise ratio (SNR) provides a convenient measure for performance, which is defined as $10\log_{10}(\sigma^2/MSE)$ decibels (dB). Figure 2.4 shows the performance limits of the first-order autoregressive Gaussian sources with $\rho = 0.9$ and $\rho = 0$. When ρ is 0.9, R_0 is 0.93. Notice that there is $-10\log_{10}(1-\rho^2) = 7.2$ dB gap between the curves for the source with memory ($\rho = 0.9$) and the memoryless Gaussian source ($\rho = 0$).

A similar fundamental limit for information transmission over a communication channel is given by Shannon's channel coding theorem. It states that there exists a number called "channel capacity" such that information can be transmitted across the channel at all rates (in bits per channel use) up to the channel capacity with arbitrarily

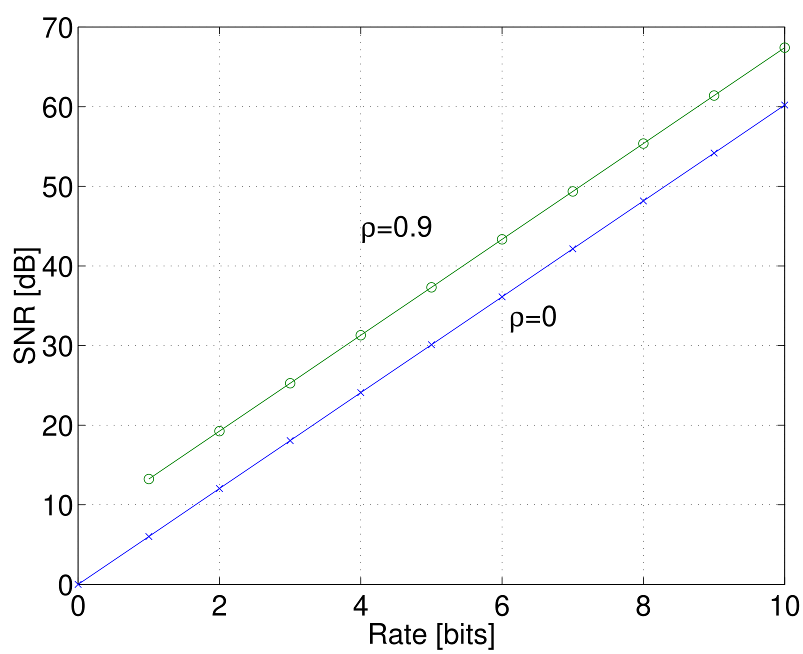


Figure 2.4: The rate-distortion curves for the first order autoregressive Gaussian Source

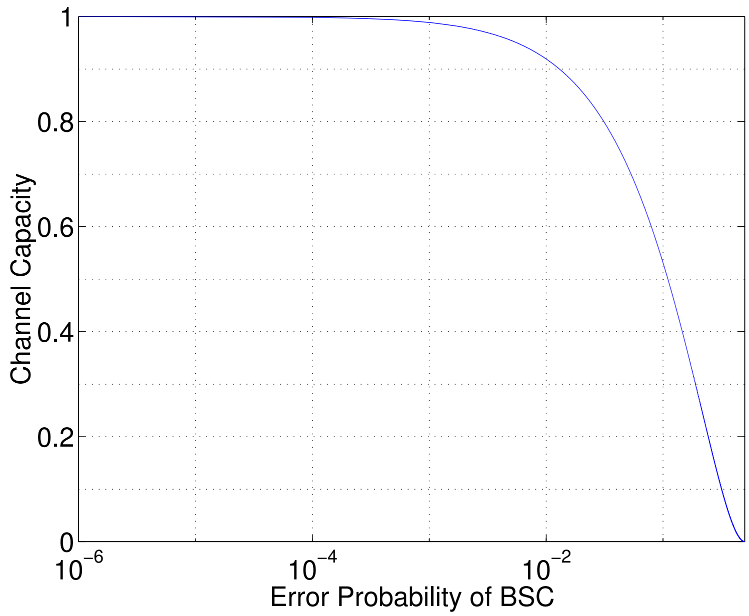


Figure 2.5. The Capacity of BSC

low error rates. Thus if the transmission rate in bits/channel use is less than the channel capacity *C*, reliable transmission is achievable over the noisy channel. For a BSC with crossover probability *p*, the channel capacity is given by

$$C = 1 + p\log_2 p + (1-p)\log_2(1-p)\,, \tag{2.2}$$

and is shown in Figure 2.5.

Shannon's separation theorem provides the theoretical performance limits for both tandem and joint source-channel coding system. Specifically, the least possible distortion is given by $D(RC)$, where *R* is channel transmission rate in channel uses per source sample and *C* is channel capacity in bits per channel uses. This distortion is achievable with a channel code with rate $R_C \cong C$ bits/channel use and a source code with rate $R_S \cong RC$ bits/source sample with no constraints on code dimension, complexity and delay. Thus, when the first-order autoregressive Gaussian source is transmitted through the cascaded source and channel coding system, the best possible performance is

$$D(RC) = \sigma^2 (1-\rho^2) 2^{-2RC}\text{, for sufficiently large } RC, \tag{2.3}$$

and is shown in Figure 2.6.

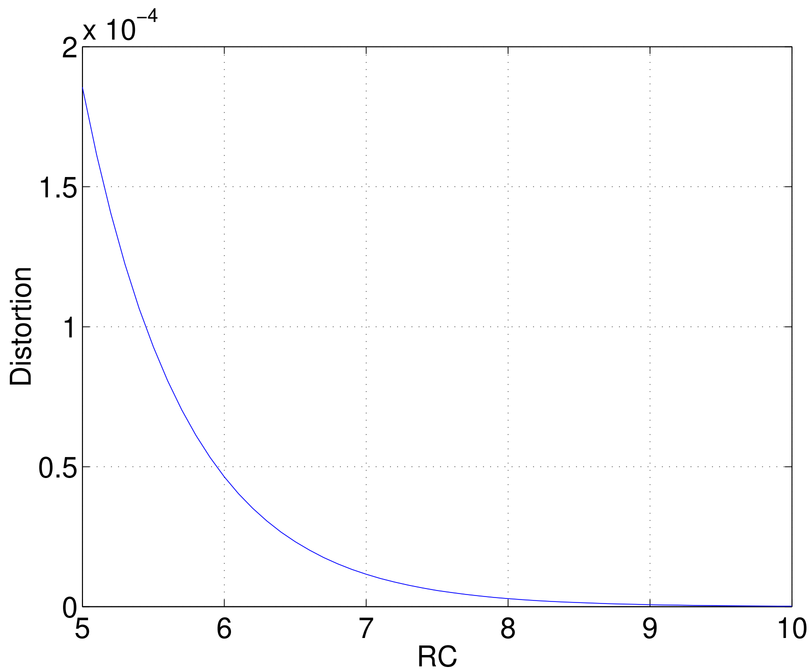


Figure 2.6. The least possible distortion of source-channel coding systems

2.3 Scalar Quantization

2.3.1 Scalar quantization for noiseless channels

Scalar Quantization is the oldest and simplest from of lossy source coding. It is a kind of first-order lossy source coding: i.e. it encodes one source sample at a time, independently of previous samples. Figure 2.7 illustrates a block diagram of a communication system containing a scalar quantizer transmitting over a binary symmetric channel. The quantizer encoder maps the source sample Y into an index set I, where the index set I is ordinarily a collection of consecutive integers from 1 to 2^n (or 0 to 2^n-1). The quantizer encoder is specified by a partition $S = \{S_i; i \in I\}$ of the source sample space, where the cell S_i is the set of source samples mapped into the index $I = i$ by the quantizer encoder.

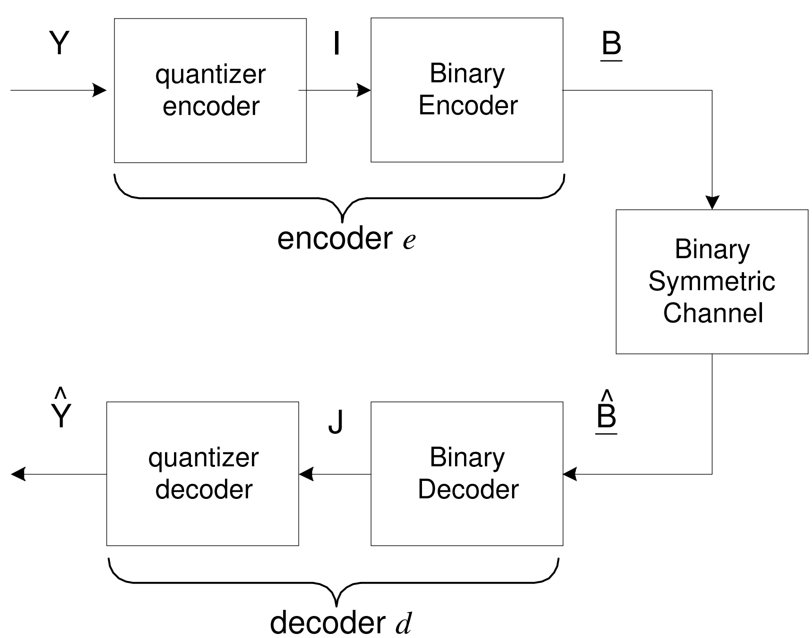


Figure 2.7: Scalar quantizer over a binary symmetric channel

The quantization index I is mapped to a binary codeword $\underline{B}$ by a binary encoder. The received sequence $\hat{\underline{B}}$ is converted to an index J, and then the quantizer decoder, or reproduction decoder, produces a reproduction $\hat{Y}$ from a codebook $C = \{\underline{c}_i; i \in I\}$ of reproduction codewords. The length of the binary codeword $\underline{B}$ may vary with the quantization index. If the binary codewords have identical lengths, the quantization is referred to as fixed-rate quantization. Otherwise it is referred to as variable-rate quantization. In fixed-rate quantization, the binary codewords are usually n-bit sequences representing the integers between 1 and 2^n (or 0 and 2^n-1). Throughout this thesis, we consider only fixed-rate quantization since variable-rate quantization is much more sensitive to channel noise and thus the performance of variable-rate quantization drops rapidly in the presence of channel errors. The number of bits in $\underline{B}$, namely n, is called the rate of quantization. The cascade of a quantizer encoder and binary encoder can be considered to be an encoder e, and the cascade of a binary decoder and quantizer decoder can be considered to be a decoder d.

For the performance measure of a scalar quantizer, mean-squared error is commonly used and is expressed by

$$D = E\left(Y - \hat{Y}\right)^2. \tag{2.4}$$

There is a well-known algorithm, Lloyd-Max algorithm [33], for designing optimal quantizers, i.e. quantizers with minimum mean-squared error. The least possible distortion $\delta_{sq,Y}(R)$ for a scalar quantizer with rate R can be tabulated for the various values of R [33]. For the high-resolution case where we assume R is large, we may use the approximation [34]

$$\delta_{sq,Y}(R) \cong \frac{1}{12}\sigma^2 \beta_1 2^{-2R}, \tag{2.5}$$

where σ^2 is the variance of the source and β_1 is the Panter-Dite factor defined as $\left(\int_{-\infty}^{\infty} p^{1/3}(x)dx\right)^3$, and where $p(x)$ is the probability density function of the source.

2.3.2 Scalar Quantization For Noisy Channels

2.3.2.1 Scalar quantizers over noisy channels

The performance of scalar quantizers designed for noiseless channels degrades rapidly in the presence of channel errors. The reason for the degradation is that the quantizers optimized with no consideration of channel errors cannot give good performance in the presence of channel errors as shown in Figure 2.8. The figure shows the performance of the quantizers (designed for noiseless channels) over noisy channels

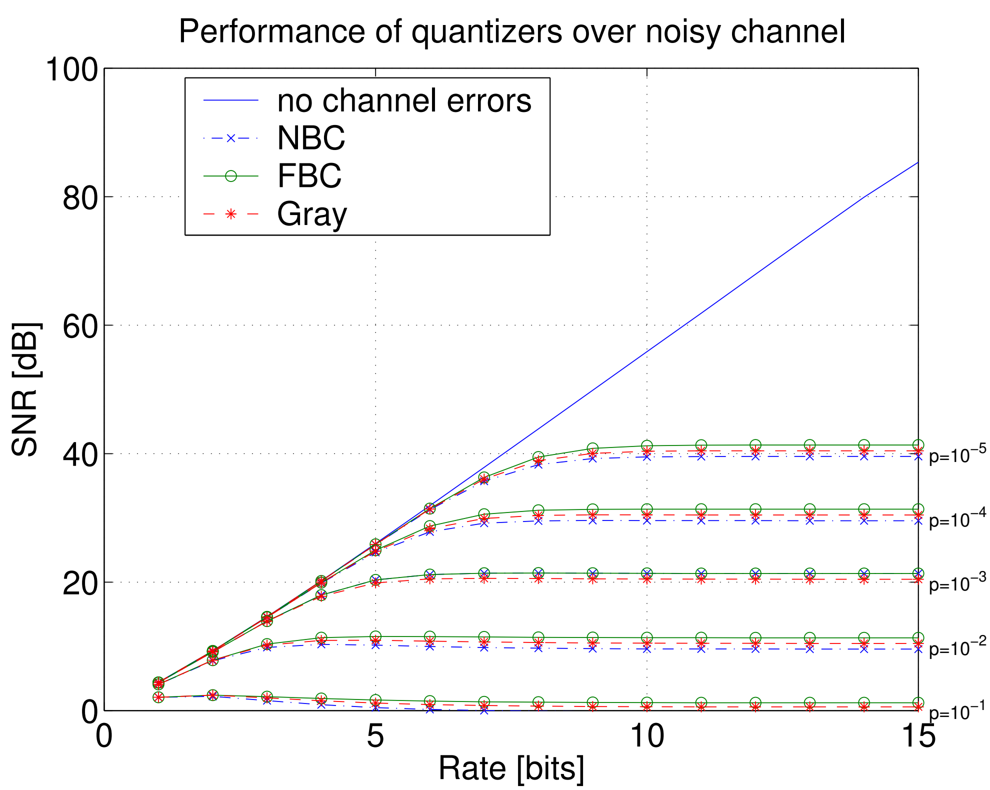


Figure 2.8: Performance of quantizers (optimized to noiseless channel) over noisy channel

using the three index assignments: the natural binary code assignment (NBC), the folded binary code Assignment (FBC) and the Gray code assignment. The index assignments refer to the codes that assign the *n*-bit binary sequences to the various codevectors, $y_1, \cdots, y_{2^n}$ of the quantizer. Table 2.1 shows examples of the three 4-bit index assignments. The performance degradation gets larger as the channel becomes noisier. Specifically, the quantizer with rate 5 loses approximately 5 dB at $p = 10^{-3}$, and 15 dB at $p = 10^{-2}$.

TABLE 2.1
Examples of 4-bit index assignments

i	NBC	FBC	Gray
0	0000	0111	0000
1	0001	0110	0001
2	0010	0101	0011
3	0011	0100	0010
4	0100	0011	0110
5	0101	0010	0111
6	0110	0001	0101
7	0111	0000	0100
8	1000	1000	1100
9	1001	1001	1101
10	1010	1010	1111
11	1011	1011	1110
12	1100	1100	1010
13	1101	1101	1011
14	1110	1110	1001
15	1111	1111	1000

2.3.2.2 Channel-optimized scalar quantization

Channel-optimized quantization [4] is the most common approach in joint source-channel coding. As shown in Figure 2.9, channel-optimized quantization consists of three components: a quantizer encoder, a binary encoder and a decoder. The quantizer encoder maps the source sample Y into an index set I, where the index set I is ordinarily a collection of consecutive integers from 1 to N (or 0 to N-1). The quantizer encoder is specified by a partition $S = \{S_i; i \in I\}$ of the source sample space, where the cell S_i is the set of source samples mapped into the index $I = i$ by the quantizer encoder. The binary encoder maps the index into an n-bit binary codeword $\underline{B}$, where $2^n \geq N$. The decoder maps a received sequence $\hat{\underline{B}}$ to a reproduction value $\hat{Y}$, where the reproduction value is among a codebook $C = \{c_1, \cdots, c_{2^n}\}$ of reproduction levels. The number of channel uses per source sample is n. The cascade of a quantizer encoder and binary encoder can be considered to be an encoder e.

In designing channel-optimized quantizers, the distortion measure is replaced by the expected distortion over the noisy channel to allow the channel statistics to be included in a quantizer design formulation. Kurtenbach and Wintz [2] introduced this

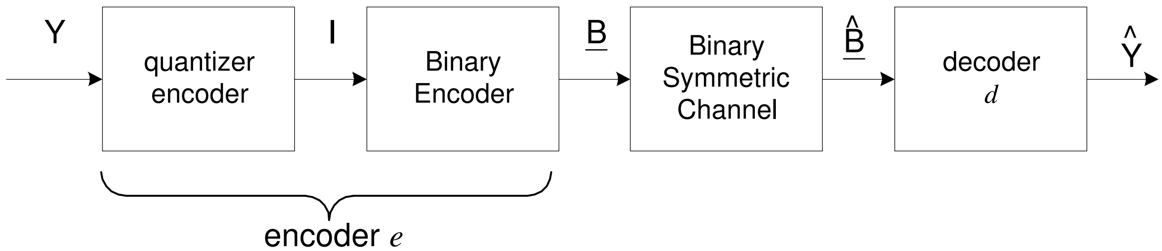


Figure 2.9: Channel-optimized quantization

method of designing scalar quantizers over noisy channels, and Farvardin and Vaishampayan [4] developed an iterative design algorithm for a channel-optimized scalar quantizer over noisy channels by providing a set of necessary conditions for the optimality of the quantizer encoder, the binary encoder and the decoder. In their design algorithm, we could encounter a situation in which $2^n > N$, i.e. the total number of quantization cells N is smaller than the total number of available codewords 2^n. Moreover, N decreases, as the channel becomes noisier. For example, Farvardin and Vaishampayan [4] reported that when the total number of available codewords is 16, N is 15 for binary symmetric error probability, p=0.005, whereas N is 6 for the more noisy channel with p=0.1. Thus, the binary encoder can be considered as an implicit channel coder, whose channel code rate is $\log_2 N \,/\, n$.

Since there may be an implicit block channel code, the approach is called "joint" because the source and channel code are designed jointly and because unlike tandem code the input blocklength of channel code is constrained to equal the output blocklength of the quantizer encoder.

2.4 Transform coding

2.4.1 Transform coding over noiseless channels

Transform coding is widely used as source coding in communication systems such as in present image and video coding standards. Figure 2.10 illustrates the structure of transform coding for noiseless channels. A block of L consecutive source samples forms an L-dimensional source vector $\underline{X} = (X_1, ..., X_L)$. By performing an orthogonal linear transform, T, the source vector is transformed to a new vector, $\underline{Y} = (Y_1, ..., Y_L)$, also with L components, often called transform coefficients, with the feature that these

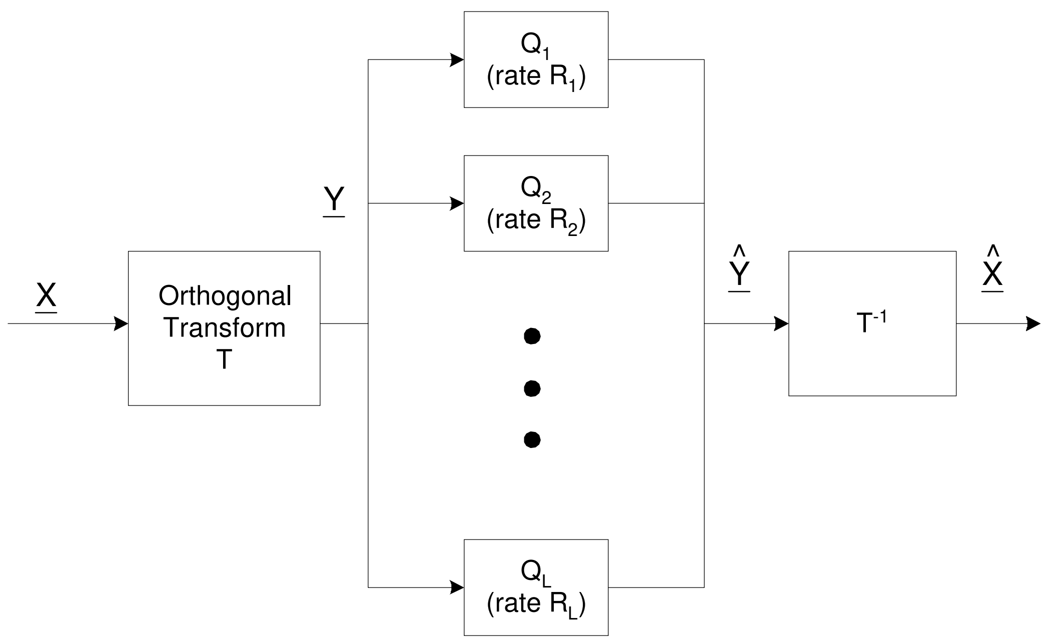


Figure 2.10: Structure of Transform Coding

coefficients are much less correlated than the original samples. In addition, the information may be much more compact in the sense of being concentrated in only a few of the transform coefficients. The *i*th transform coefficient is quantized by a scalar quantizer Q_i with rate R_i, where Q_i consists of an encoder *e* and a decoder *d* as in Figure 2.7. The reconstructed approximation to the original vector, $\underline{\hat{X}} = (\hat{X}_1, \cdots, \hat{X}_L)$, is then obtained by performing the corresponding inverse transform on the quantized transform vector $\underline{\hat{Y}}$. The rate of transform code is given by

$$R = \frac{1}{L}\sum_{i=1}^{L} R_i \,, \tag{2.6}$$

where R_i is the rate of *i*th quantizer. The most convenient distortion measure to use for assessing performance of this coding scheme is the overall mean squared error given by

$$D = \frac{1}{L}\sum_{i=1}^{L} E\left(X_i - \hat{X}_i\right)^2 = \frac{1}{L}E\left\|\underline{X} - \underline{\hat{X}}\right\|^2, \tag{2.7}$$

where $\left\|\underline{X} - \underline{\hat{X}}\right\|$ is defined as

$$\left\|\underline{X} - \underline{\hat{X}}\right\| = \sqrt{\sum_{i=1}^{L}\left(X_i - \hat{X}_i\right)^2}\ .$$

The orthogonality of the transform implies

$$\begin{aligned} D &= \frac{1}{L}E\left\|\underline{X} - \underline{\hat{X}}\right\|^2 = \frac{1}{L}E\left\|\underline{Y} - \underline{\hat{Y}}\right\|^2 \\ &= \frac{1}{L}\sum_{i=1}^{L} E\left(Y_i - \hat{Y}_i\right)^2 = \frac{1}{L}\sum_{i=1}^{L} D_{Y_i,Q_i}, \end{aligned} \tag{2.8}$$

where D_{Y_i,Q_i} is the distortion of the ith quantizer Q_i applied to U_i.

Given the transform dimension L and a fixed rate R, to find the least possible distortion, three optimization procedures are needed: (1) given $R_1,\ldots, R_L$ satisfying (2.6) and the transform T, choose the quantizers, $Q_1, \ldots, Q_L$ to have rates $R_1,\ldots, R_L$, respectively to minimize distortion, (2) choose $R_1,\ldots, R_L$ to minimize distortion assuming $Q_1, \ldots, Q_L$ are chosen as in (1) and $\frac{1}{L}\sum_{i=1}^{L} R_i \le R$, and (3) choose transform T to minimize distortion assuming $R_1,\ldots, R_L$ and $Q_1, \ldots, Q_L$ are chosen as in (1) and (2).

Given $R_1,\ldots,R_L$ satisfying (2.6) and T, we should choose Q_i to minimize D_{Y_i,Q_i} subject to the constraint that its rate is R_i. Then $D_{Y_i,Q_i} = \delta_{sq,Y_i}(R_i)$, where $\delta_{sq,Y_i}(R_i)$ is the least possible distortion for scalar quantization of Y_i. Thus, we have

$$D = \frac{1}{L}\sum_{i=1}^{L}\delta_{sq,Y_i}(R_i) \tag{2.9}$$

Using (2.5) for $\delta_{sq,Y_i}(R_i)$, we now have

$$D \cong \frac{1}{L}\sum_{i=1}^{L}\frac{1}{12}\sigma_i^2\beta_1 2^{-2R_i}\ . \tag{2.10}$$

The problem of finding optimal rates $R_1,\ldots, R_L$ that minimize the above expression for distortion subject to the rate constraint (2.6) is called a bit or rate allocation, i.e. an allocation of *LR* bits among the individual coefficients. A direct method of finding an optimal bit allocation would find a rate distribution that minimizes the overall distortion among all possible distributions. However, this method has a drawback in practice because the computational burden is huge. Thus, many numerical algorithms for minimizing (2.10) have been developed to overcome this drawback [32, pp. 225 –235]. However, these numerical algorithms require the optimal distortion values for the various rates. In some cases, the optimal distortion values are not known exactly.

Another approach to finding an optimal bit allocation is a method of finding explicit expressions for the optimal distortion using high-resolution quantization approximations and minimizing the resulting distortion expression with familiar Lagrange techniques of calculus or common inequalities [32]. High-resolution analysis gives the optimal rate allocation:

$$R_i = R + \frac{1}{2}\log\frac{\sigma_i^2}{\left(\prod_{i=1}^{L}\sigma_i^2\right)^{\frac{1}{L}}}. \tag{2.11}$$

Using the optimal bit allocation given by (2.11), the least possible distortion is expressed as

$$D(R) \cong \frac{\beta_1}{12}\left(\prod_{i=1}^{L}\sigma_i^2\right)^{\frac{1}{L}} 2^{-2R}. \tag{2.12}$$

It is known that the Karhunen-Loeve transform (KLT) is the optimal transform for Gaussian sources in sense that the KLT minimizes the distortion given by (2.12) [35]. When the Karhunen-Loeve transform code is used as the transform code, the least possible distortion for a Gaussian source with autocorrelation matrix *K* is

$$\delta_{tr}(R) = \frac{1}{12}\beta_1 |K|^{\frac{1}{L}} 2^{-2R}, \quad \text{for large } R \tag{2.13}$$

where L is the transform size and $\beta_1 = 2\pi 3^{3/2}$ for Gaussian source. Compared to the distortion-rate function given by (2.1), the KL transform coding loses about 4.34 dB.

2.4.2 Transform coding over noisy channels

In this section, we review two approaches to combat channel errors in transform coding systems. These approaches, which use transform coding as the source code, are specific examples of the two common strategies: joint and tandem source-channel coding, described in Section 2.1.

2.4.2.1 Transform coding in tandem source-channel coding

When a transform code is used over a noisy channel, the coefficients are usually protected by a channel code to combat the noise as shown as in Figure 2.11, where each encoder e_i and decoder d_i are the encoder and decoder for a scalar quantizer as in Figure 2.7. For the KL transform code followed by a channel code over a BSC with channel capacity C and transmission rate R, the least possible distortion is given by

$$\delta_{tr}(RC) = \frac{1}{12}\beta_1 |K|^{\frac{1}{L}} 2^{-2RC}, \text{ for large } RC. \tag{2.14}$$

Figure 2.12 shows the best possible performance of ideal source and channel coding systems, and of the conventional KL transform coding with ideal channel coding, described by (2.3) and (2.14) respectively. We can see the 4.3 dB loss of KL transform coding and ideal channel coding as mentioned in the previous section.

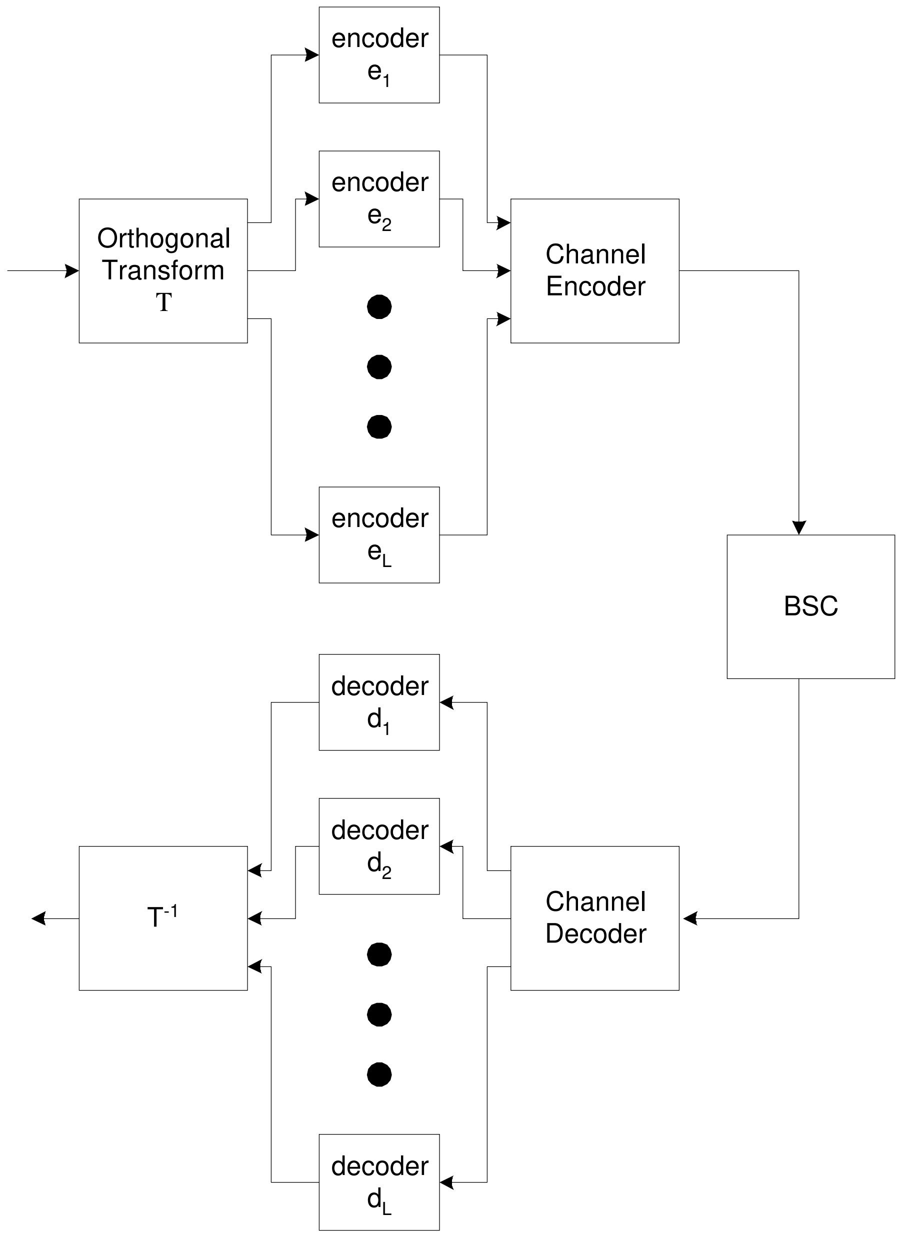


Figure 2.11: The structure of transform coding in tandem source-channel coding

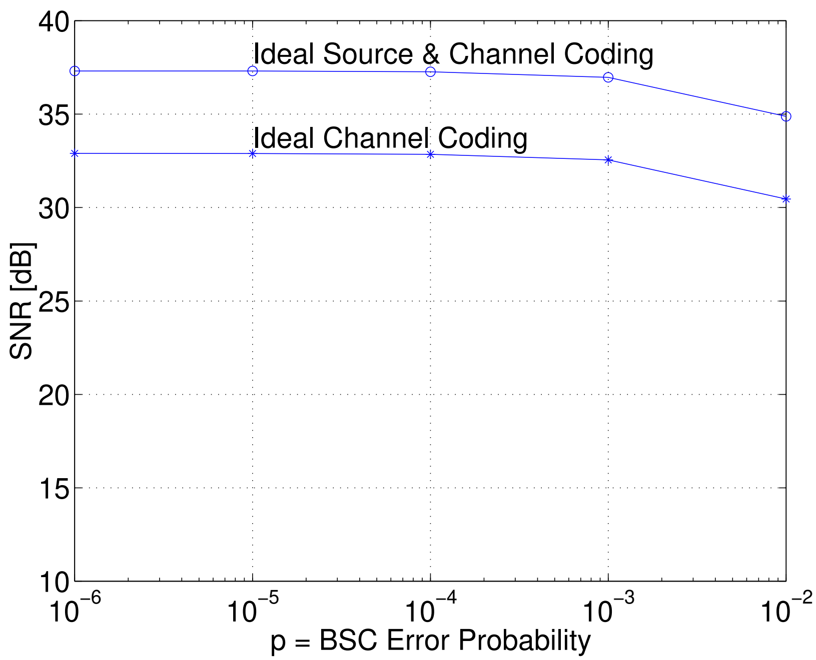


Figure 2.12: The best possible performance of ideal source and channel coding system, and KL transform and ideal channel coding system for the first order autoregressive Gaussian Source with $\rho = 0.9$. The transform size of KLT is L=128 and the transmission rate is R=5.

2.4.2.2 Transform coding in joint source-channel coding: Channel-optimized transform coding

The most common and interesting example of transform coding in the joint source-channel coding strategy is channel-optimized transform coding [5]. A block diagram of channel-optimized transform coding is illustrated in Figure 2.13. Each encoder and decoder are the encoder and decoder respectively of a channel-optimized quantization as in Figure 2.9. For an integer L to be specified, an L-dimensional transform T maps the first L source data samples $\underline{U} = (U_1, U_2, \cdots, U_L)$ into an L-dimensional vector of coefficients $\underline{V} = T\underline{U}$. Each coefficient V_i is then scalar quantized with a quantizer Q_i having $N_{S,i} = 2^{R_{S,i}}$ levels denoted $y_{i,1} < \cdots < y_{N_{S,i}}$. The distortion of the quantizer Q_i is $D_{S,i} = E[V_i - Q_i(V_i)]^2$, where $Q_i(V_i)$ is expressed as $Q_i(V_i)$ = $d(\mathrm{e}(V_i))$ by the encoder e and the decoder d in Figure 2.9. Each quantization level $y_{i,j}$ is

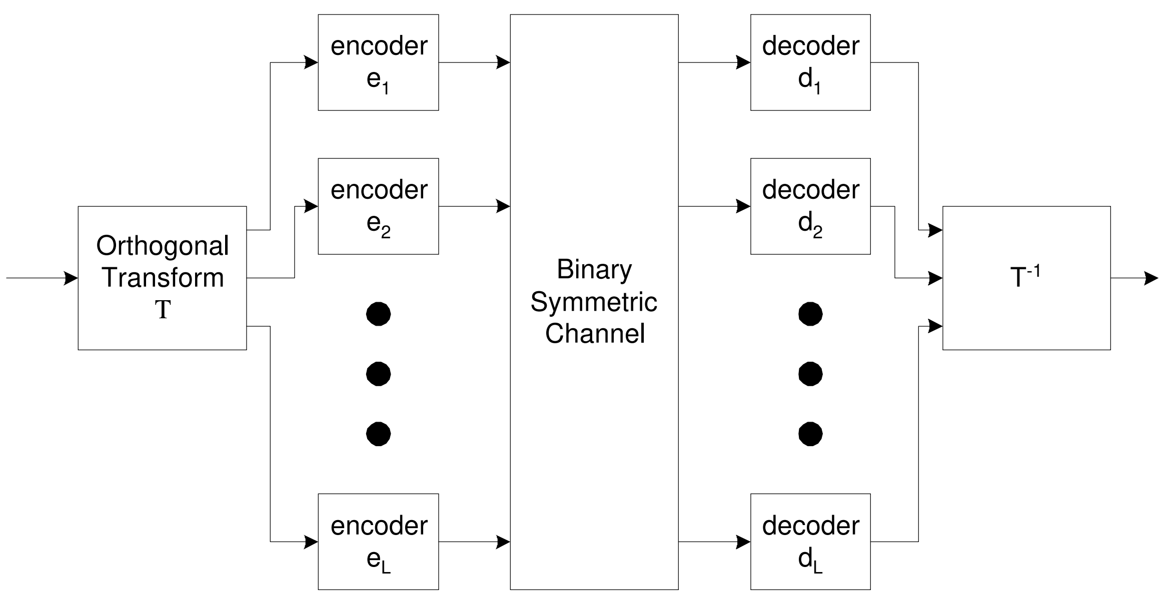


Figure 2.13: Block diagram of the channel-optimized transform coder

assigned an $R_{S,i}$-bit binary codeword when V_i is quantized to $y_{i,j}$. Let $\underline{B}_i$ denote the binary sequence produced when quantizing V_i. The resulting vector of binary sequences $\underline{B} = (\underline{B}_1, \underline{B}_2, \cdots, \underline{B}_L)$ is transmitted across a BSC. The received sequence $\hat{\underline{B}}$, each component of which is a binary sequence, is decoded component-wise by a decoder, and the resulting vector $\hat{\underline{V}}$ is then transformed by T^{-1}. The overall mean-squared error distortion of the transform coder is given by

$$D = \frac{1}{L} \sum_{i=1}^{L} D_{S,i}(R_{S,i}), \tag{2.15}$$

and the overall coding rate is given by

$$\overline{R} = \frac{1}{L} \sum_{i=1}^{L} R_{S,i} \text{ bits/sample.} \tag{2.16}$$

For a given transform *T*, Vaishampayan and Farvardin [5] formulated an optimization problem in which the mean-squared error of the transform coding over the noisy channel is minimized subject to a constraint on the overall coding rate. They showed that the optimization problem is reduced to the following.

1) The optimal allocation of bits for encoding the transform coefficients in such a way that both the source and the channel characteristics are taken into consideration.

2) The design of optimal scalar quantizers for the transmission of each of the transform coefficients over a noisy channel.

Thus, when R channel uses are available per source symbol, their channel-optimized transform coder chooses the rate allocations $R_{S,i}$, the quantization levels $y_{i,j}$ and the binary sequences $\underline{w}_{i,j}$ to minimize the overall mean-squared error distortion given by (2.14) subject to the constraint that the overall coding rate is at most *R*. This optimization method is described in [5] for a binary symmetric channel (parameterized by *p*).

CHAPTER III

JOINT AND TANDEM SOURCE-CHANNEL CODING WITH COMPLEXITY CONSTRAINTS

In this chapter, two common source-channel coding strategies, joint and tandem source-channel coding, are compared on the basis of distortion vs. complexity by analyzing specific representatives of each. The representative systems are introduced, and for each system, formulas for the mean-squared error and complexity are derived. Then an optimization procedure which minimizes distortion subject to a complexity constraint is described. The results that followed compare both systems on the basis of distortion vs. complexity.

3.1 Introduction

As discussed earlier, ever since Shannon showed that one can optimally communicate analog data across a noisy channel using tandem source-channel coding, in which a source code designed without regard to the possibility of channel errors is followed by a channel code designed without regard to the source, much effort has been devoted to two separate areas in designing a communication system: designing source codes under ideal channel conditions and designing channel codes without any consideration of the source characteristics. Thus, many communication systems have been based on the tandem connection of such independent effort. However, Shannon showed how optimal performance could be obtained without regard to complexity and delay. For practical systems having limitations on complexity and delay, there has been considerable interest in joint source-channel coding, in which source and channel code are jointly designed to combat channel errors. One common example of the joint source-channel coding is channel-optimized quantization [5], in which no explicit channel code is used. The motivation for such joint source-channel coding is that it can obtain better performance with less complexity and/or delay than tandem source-channel coding. On the other hand, little quantitative evidence for this claim has appeared in the literature. In this chapter, we seek such evidence by finding whether joint source-channel coding achieves better performance with less complexity than tandem source-channel coding by analyzing specific representative systems. The comparison of joint and tandem source-channel coding with delay constraints is postponed to Chapter V.

To avoid idiosyncrasies, we examine systems that are as “plain vanilla” as possible. Specifically, as the joint source-channel code, we consider the channel-optimized transform code studied by Vaishampayan and Farvardin [5] and described in Chapter II. As the tandem source-channel code, we consider a conventional transform code (optimized for a noiseless channel) followed by a Reed-Solomon (RS) channel code. We do not use entropy coding with either system, because its performance tends to be highly method specific, making it unlikely that one could choose a representative plain vanilla version.

In the context of complexity, which is measured by the number of arithmetic operations per source sample, tandem coding requires allocation of complexity between source and channel codes, while channel-optimized quantization does not require allocation of complexity, because it doesn’t have an explicit channel code. Thus the main interest in tandem coding is how the computational complexity is divided between source and channel codes in order to get the best possible performance under a given complexity constraint. If we allocate all of the complexity to the source code, we have a conventional source coding system. This source coding system does not have good performance over noisy channels because it is designed under noiseless channel conditions. To get better performance over noisy channels, the source code should be optimized for the noisy channels. So channel-optimized quantization is considered as a system in which no explicit channel code exists, and so the source code has 100% of the given computational complexity.

We choose mean-squared error (MSE) as the fidelity measure which, despite its faults, is by far the most commonly used. We evaluate the performance of these systems on Gauss-Markov (first-order AR) source over binary symmetric channels (BSC). The combination of MSE and Gauss-Markov source provides a level playing field that is unlikely to reward one system over an another due to idiosyncrasies. They also provide tractability, meaning that performance can be computed, rather than needing to be measured from a simulation. Finally, the complexities of the two systems are estimated by counting the numbers of arithmetic operations per data sample required for encoding and decoding.

An outline of this chapter is as follows, Section 2 introduces the representative systems under consideration, Section 3 develops formulas for distortion, Section 4 develops expressions for complexity, Section 5 describes the optimization under a complexity constraint, and Section 6 presents the results and concluding remarks.

3.2 The Systems

In this section, we describe the representative systems mentioned previously. As the data source model, we choose a one-dimensional, stationary Gauss-Markov random sequence denoted $\{U_i\}$, with zero mean, unit variance, and correlation coefficient $\rho = E(U_i U_{i+1})$. As the channel model, we choose a binary symmetric channel (stationary and memoryless) whose error probability is denoted p. The number of channel uses available to transmit one source sample is denoted R .

3.2.1 Joint source-channel coding system

As the joint source-channel coding strategy, we use the channel-optimized transform code described in Section 2.3.2.2. Our representative system for channel-optimized transform coding uses the optimal transform for Gaussian sources, the Karhunen-Loeve transform, as the transform.

3.2.2 Tandem source-channel coding system

The tandem source-channel coding strategy uses the conventional Karhunen-Loeve transform source code (optimized for a noiseless channel) followed by a Reed-Solomon channel code, as shown in Figure 3.1. The rate allocation and quantization levels of the transform code in the tandem system are optimized for the source by conventional means without regard to the channel.

In the tandem source-channel coding system, the source code operates as follows. For a positive integer L to be specified, an L-dimensional Karhunen-Loeve transform T maps the first L source samples $\underline{U} = (U_1, U_2, \cdots, U_L)$ into an L-dimensional vector of coefficients $\underline{V} = T\underline{U}$. Each coefficient V_i , which is itself Gaussian with variance denoted σ_i^2 , is then scalar quantized with a nonuniform fixed-rate, Lloyd-Max quantizer Q_i with $N_{s,i} = 2^{R_{S,i}}$ levels denoted $y_{i,1} < \cdots < y_{i,N_{s,i}}$. That is, Q_i is designed to have the least MSE, $D_{S,i}(R_{S,i})$, when quantizing V_i with $N_{S,i}$ levels and no channel errors. To each quantization level is associated an $R_{S,i}$-bit binary sequence via the natural binary assignment. That is, $y_{i,j}$ is assigned the $R_{S,i}$-bit binary sequence $\underline{w}_{i,j}$ that represents the integer $j-1$. Let $\underline{B}_i$ denote the binary sequence produced when quantizing V_i .

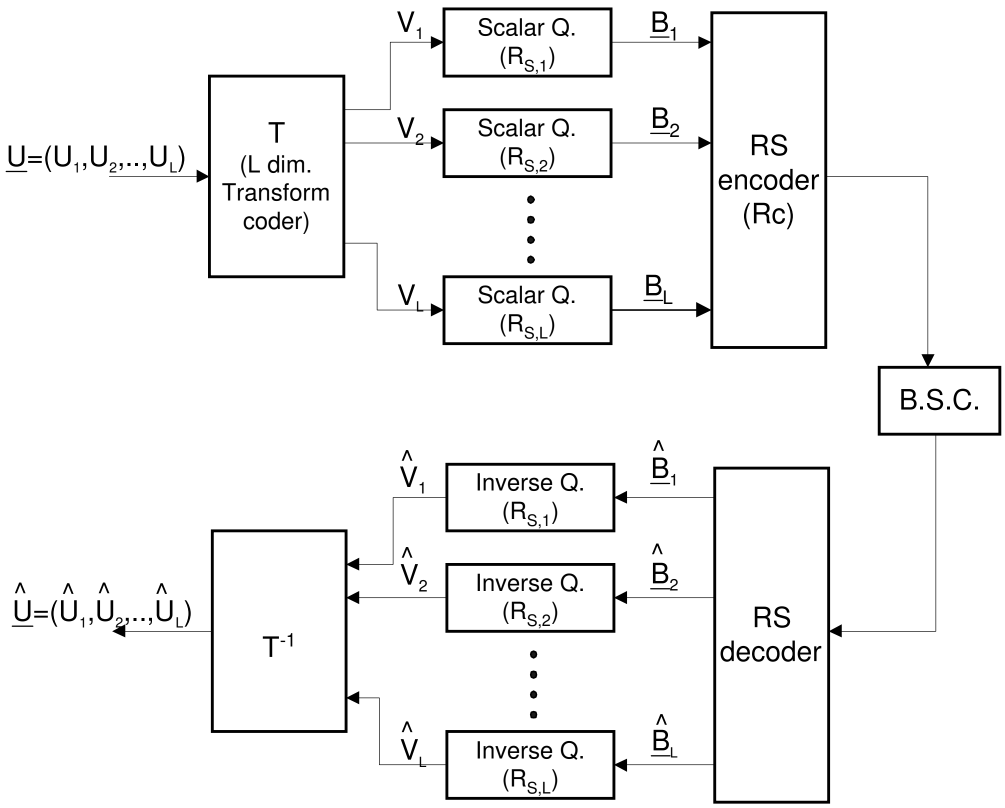


Figure 3.1: The block diagram of the tandem source-channel coding system

Subsequent blocks of L source data samples, such as $U_{L+1},\cdots,U_{2L}$, are encoded in the same way, i.e. with the same transform and the same quantizers.

The overall source coding rate of tandem system is given by

$$R_S = \frac{1}{L}\sum_{i=1}^{L} R_{S,i} \text{ bits/sample,} \tag{3.3}$$

which is the average of the individual quantizer rates. When a target source coding rate R_S is specified, a conventional bit allocation procedure is used to choose the $R_{S,i}$'s to be integers that sum to $R_S L$ and minimize the MSE, given by

$$D_S = \frac{1}{L}\sum_{i=1}^{L} D_{S,i}(R_{S,i}), \tag{3.4}$$

which is the average of the individual scalar quantizer distortions when no channel errors are present.

The channel code is chosen to be an (*n,k,m*) Reed-Solomon (R-S) code with $n = 2^m - 1$, $k \le n$ and channel coding rate denoted $R_C = k/n$. With such a code, *k* data symbols are encoded into *n* channel symbols, where each data symbol consists of *m* bits from the transform code and each channel symbol consists of *m* bits used as the input to the BSC. The minimum distance of such a code is *n*-*k*+1, which means that its decoder can correct any pattern of $t = \lfloor (n-k)/2 \rfloor$ or fewer channel symbol errors, where a channel symbol error is said to occur whenever the channel introduces one or more bit errors within an *m*-bit channel symbol.

The R-S encoder is chosen to be systematic in sense that each codeword has the *k* data symbols that it encodes as its first *k* channel symbols. When the channel rate *R* is specified, it is required that the source and channel codes be chosen so that $R_S / R_C = R$. We do not assume that the number of bits, LR_S, produced by the source code is equal to, a divisor of, or a multiple of the number of bits, *km*, encoded by one R-S codeword. Rather the bits produced by the transform code are concatenated into a single stream that is parsed into blocks of *km* for encoding by the R-S encoder. It follows that the bits $\underline{B}_i$ describing coefficient V_i might be embedded in two or more symbols from one or two codewords.

The R-S decoder operates as follows. When the BSC output, considered as an *n*-symbol sequence, is within Hamming distance *t* of some R-S codeword, the decoder produces the first *k* symbols of that codeword, where the Hamming distance between two sequences is defined to be the number of bits in which they differ. On the other hand, when the Hamming distance between the BSC output and each codeword exceeds *t*, the received sequence and decoder are said to FAIL, and the decoder simply produces the first k channel output symbols. The resulting decoded symbol stream is parsed into blocks of LR_S bits and presented to the transform decoder. We have found that using this strategy when the received sequences FAIL's works better than either simply producing the first k symbols of the closest codeword, or declaring an ERASURE and replacing the missing transform coefficients with zeroes. The possibility of exploiting the correlation between successive transform vectors was not used because of its potential complexity and because the results obtained might be too specific to the chosen source model.

3.3 Distortion

3.3.1 Channel-optimized transform code

The distortion of the channel-optimized transform coding may be computed by the method of [4].

3.3.2 Tandem source-channel code

The tandem source-channel coding system is specified by a source parameter ρ, channel parameters p and R, source code parameters L and R_S, and channel code parameters n, k, and m. . We consider an infinite number of source samples and view the output of the transform coder as an infinite sequence $V_1, V_2, \cdots, V_L, V_{L+1}, \cdots, V_{2L}, V_{2L+1}, \cdots$, $V_{3L}, \cdots$, where V_{jL+i} is the ith coefficient of the jth transform. The V_{jL+i}'s for different j's are quantized by the same quantizer, Q_i. The computation of distortion is complicated by the fact that the binary sequence $\underline{B}_i$ representing coefficient V_i is, in general, embedded in a different way in the channel coded bit stream than the bits $\underline{B}_{L+i}$ representing coefficient V_{L+i} in the second transform block, even though V_{L+i} is quantized in the same way as V_i. For example, $\underline{B}_1$ will be the first $R_{S,1}$ bits of the first channel codeword, but depending on the values, R_S , L, k and m, $\underline{B}_{L+1}$ might appear later in the same codeword or might be divided between the first and second codeword, to mention just two of the possibilities. Moreover, the bits of $\underline{B}_1$ would begin a symbol, whereas those of $\underline{B}_{L+1}$ might not.

In effect we need to average the MSE of coefficients of the same type as V_i over the various ways that the bits representing such coefficients can be embedded in the channel coded bit stream. However, one can easily see that $\underline{B}_i$, $\underline{B}_{M_S L+i}$, $\underline{B}_{2M_s L+i}$, ... are embedded in the same way in the channel coded bit stream, where M_S is a positive integer such that $M_S LR_S = M_C km$, where M_S and M_C are relatively prime. That is, M_S is the smallest integer such that the bits produced by M_S applications of the transform code can be packed into an integer number of R-S codewords. $M_C = M_S LR_S / km$ is the number of channel codewords used to encode the M_S data vectors. This relationship is illustrated in Figure 3.2. Thus, the distortion D_i for the ith source sample, $E(V_i - \hat{V}_i)^2$, is periodic with period $M_S L$. Since our transform is orthogonal, i.e. $T^t = T^{-1}$, the average distortion can be expressed as

$$\begin{aligned} D &= \frac{1}{M_S L}\sum_{i=1}^{M_S L} E(U_i - \hat{U}_i)^2 \\ &= \frac{1}{M_S L}\sum_{i=1}^{M_S L} E(V_i - \hat{V}_i)^2 \quad . \\ &= \frac{1}{M_S L}\sum_{i=1}^{M_S L} D_i \end{aligned} \tag{3.5}$$

To compute D_i's we make the assumption that when the decoder does not FAIL, it always produces the correct decision. As can be seen from [29], this is a good assumption when the error correcting capability t is at least 4. With this assumption, D_i can be

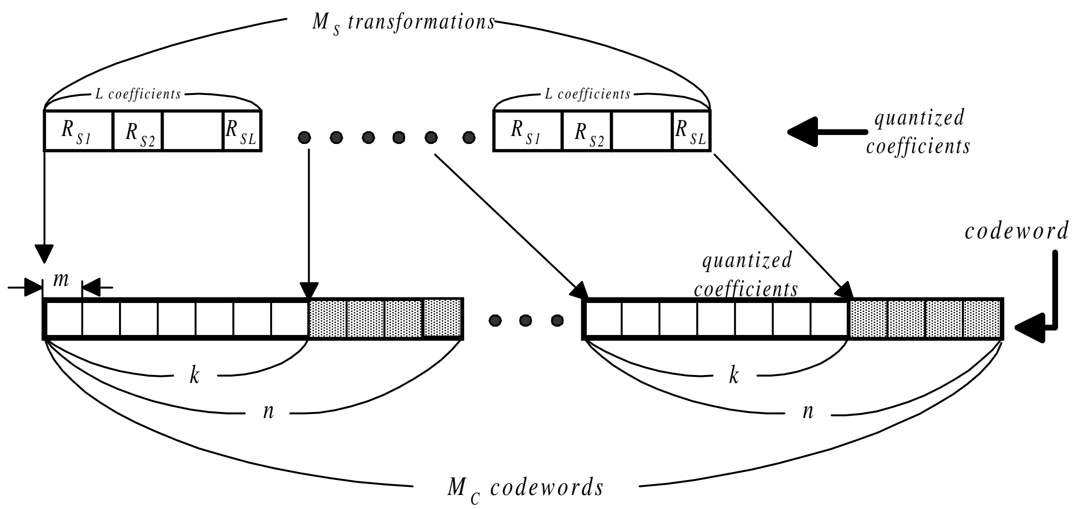


Figure 3.2: $M_S L$ data samples are encoded and parsed into M_C codewords

expressed as the sum of conditional distortions:

$$D_i \cong E[(V_i - \hat{V}_i)^2 \mid NF_i]P(NF_i) + E[(V_i - \hat{V}_i)^2 \mid F_i]P(F_i), \tag{3.6}$$

where F_i is the event that the received sequence containing the bits $\underline{B}_i$ representing V_i FAILS, or if $\underline{B}_i$ is spread over two received sequences, that one or both of these FAILS, and where NF_i is the NO FAIL event, i.e. the complement of F_i. Since the distortion given NF_i is the distortion of the source transform code when no channel errors are present, (3.6) becomes

$$D_i = D_{S,i}(R_{S,i})P(NF_i) + E[(V_i - \hat{V}_i)^2 \mid F_i]P(F_i). \tag{3.7}$$

By the usual argument of combined source-channel coding (cf. [27], pp. 179-181),

$$E[(V_i - \hat{V}_i)^2 \mid F_i] = D_{S,i} + D_{C,i}, \tag{3.8}$$

where $D_{S,i} = D_{S,i}(R_{S,i})$ is the quantizer distortion described earlier and $D_{C,i}$ is the channel distortion (assuming F_i) defined by

$$D_{C,i} = \sum_{j,j'=1}^{N_{S,i}} (y_{i,j} - y_{i,j'})^2 P_i(j) P_i(\underline{w}_{i,j'} | \underline{w}_{i,j}, F_i), \tag{3.9}$$

where $P_i(j)$ is the probability that the quantizer for V_i produces the *j*th level $y_{i,j}$ and $P_i(\underline{w}_{i,j'} \mid \underline{w}_{i,j,} F_i)$ is the probability that the channel decoder puts out the binary sequence for $y_{i,j'}$ given that the channel input was the binary sequence for $y_{i,j}$ and that the sequence(s) containing $\underline{B}_i$ FAILED. Combining (3.5), (3.7)-(3.8) gives

$$D = D_S + D_C \ , \tag{3.10}$$

where D_S is the distortion of the transform code with no channel errors, as defined by (3.4) in the previous section, and D_C is the channel distortion (overall) defined by

$$\begin{aligned} D_C &= \frac{1}{M_S L} \sum_{i=1}^{M_S L} D_{C,i} P(F_i) \\ &= \frac{1}{M_S L} \sum_{i=1}^{M_S L} \sum_{j=1}^{N_{S,i}} \sum_{j'=1}^{N_{S,i}} (y_{i,j} - y_{i,j'})^2 P_i(j) P_i(\underline{w}_{i,j'} | \underline{w}_{i,j}, F_i) P(F_i). \end{aligned} \tag{3.11}$$

In (3.11), the terms $(y_{i,j} - y_{i,j'})^2$ and $P_i(j)$ are easily computed. The only difficulty is in computing the term $P_i(\underline{w}_{i,j'} \mid \underline{w}_{i,j,} F_i)\ P(F_i)$. However, the term $P(F_i)$ is cancelled out in expressing $P_i(\underline{w}_{i,j'} \mid \underline{w}_{i,j,} F_i)$, so we focus on finding an expression for $P_i(\underline{w}_{i,j'} \mid \underline{w}_{i,j,} F_i)$. To find an expression for $P_i(\underline{w}_{i,j'} \mid \underline{w}_{i,j,} F_i)$, one must take into account the position of $\underline{B}_i$ in the encoded bit stream relative to symbol and codeword boundaries. Thus we consider two cases in which $\underline{B}_i$ is entirely contained within one R-S codeword and $\underline{B}_i$ is spread over two R-S codewords.

A. The case that $\underline{B}_i$ is entirely contained within one R-S codeword

Consider the case that $\underline{B}_i$ is entirely contained within one R-S codeword (rather than two) as shown in Figure 3.3. Let $\underline{\tilde{B}}_i$ denote the raw channel output in response to $\underline{B}_i$, and let $\underline{\hat{B}}_i$ denote the corresponding decoder output. Since with our decoder strategy $\underline{\hat{B}}_i = \underline{\tilde{B}}_i$ when the decoder FAILS, we have

$$\begin{aligned} P_i(\underline{w}_{i,j'} \mid \underline{w}_{i,j}, F_i) &= \Pr(\underline{\hat{B}}_i = \underline{w}_{i,j'} \mid \underline{B}_i = \underline{w}_{i,j}, F_i) \\ &= \Pr(\underline{\tilde{B}}_i = \underline{w}_{i,j'} \mid \underline{B}_i = \underline{w}_{i,j}, F_i) \\ &= \frac{P_i(j)\Pr(\underline{\tilde{B}}_i = \underline{w}_{i,j'} \mid \underline{B}_i = \underline{w}_{i,j}) P(F_i \mid \underline{B}_i = \underline{w}_{i,j}, \underline{\tilde{B}}_i = \underline{w}_{i,j'})}{P_i(j)P(F_i)} \\ &= \frac{\Pr(\underline{\tilde{B}}_i = \underline{w}_{i,j'} \mid \underline{B}_i = \underline{w}_{i,j})}{P(F_i)} P(F_i \mid \underline{B}_i = \underline{w}_{i,j}, \underline{\tilde{B}}_i = \underline{w}_{i,j'}), \end{aligned} \tag{3.15}$$

where

$$\Pr(\underline{\tilde{B}}_i = w_{i,j'} \mid \underline{B}_i = w_{i,j}) = p^{d_H(w_{i,j}, w_{i,j'})}(1-p)^{R_{S,i} - d_H(w_{i,j}, w_{i,j'})}, \tag{3.16}$$

and $d_H(\underline{w}_{i,j}, \underline{w}_{i,j'})$ denotes the Hamming distance between the binary sequences

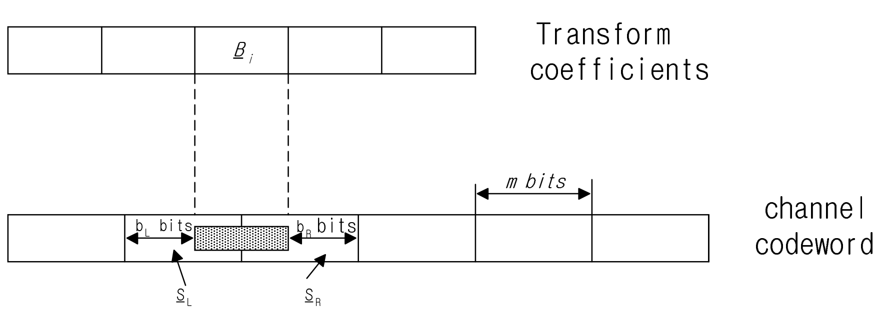


Figure 3.3: The case that $\underline{B}_i$ is entirely contained within one R-S codeword

representing $\underline{w}_{i,j}$ and $\underline{w}_{i,j'}$. Inserting (3.15) into (3.11), the channel distortion due to *i*th transform coefficient is expressed as

$$D_{C,i} = \frac{1}{P(F_i)} \sum_{j=1}^{N_{S,i}} \sum_{j'=1}^{N_{S,i}} \{(y_{i,j} - y_{i,j'})^2 P_i(j) \bullet \Pr(\underline{\tilde{B}}_i = \underline{w}_{i,j'} \mid \underline{B}_i = \underline{w}_{i,j}) P(F_i \mid \underline{B}_i = \underline{w}_{i,j}, \underline{\tilde{B}}_i = \underline{w}_{i,j'})\} \tag{3.17}$$

Now, to derive an expression for the only unknown term in channel distortion equation (3.17), $P(F_i \mid \underline{B}_i = \underline{w}_{i,j}, \underline{\tilde{B}}_i = \underline{w}_{i,j'})$, let n_t denote the number of channel symbols containing at least one bits of $\underline{B}_i$, let $\underline{S}_L$ $(\underline{S}_R)$ denote the channel symbol containing the left-most (right-most) bit of $\underline{B}_i$, let b_L (b_R) denote the number of bits of $\underline{B}_i$ contained in $\underline{S}_L$ $(\underline{S}_R)$, and let n_e be the number of channel symbol errors caused by the bit errors between $\underline{B}_i$ and the channel output $\underline{\tilde{B}}_i$, i.e. by the bit errors in the position occupied by $\underline{B}_i$ (See Figure 3.3). Then the probability of one or more errors in the bits of $\underline{S}_L$ $(\underline{S}_R)$ that are not occupied by $\underline{B}_i$ is

$$P_{sy,L} = 1 - (1-p)^{m-b_L}, \quad (P_{sy,R} = 1 - (1-p)^{m-b_R}). \tag{3.18}$$

The probability $P(F_i \mid \underline{B}_i = \underline{w}_{i,j}, \underline{\tilde{B}}_i = \underline{w}_{i,j'})$ should be considered under 4 different cases, depending on whether cause or do not cause symbol errors in bit errors between $\underline{B}_i$ and the channel output $\underline{\tilde{B}}_i$ in $\underline{S}_L$ and $\underline{S}_R$.

Case 1: First, consider the case that both $\underline{S}_L$ and $\underline{S}_R$ have bit errors caused by the bit errors between $\underline{B}_i$ and the channel output $\underline{\tilde{B}}_i$. For this case, the decoder declares FAIL if there are more than $t+1-n_e$ errors in the n - n_t symbols not occupied by $\underline{B}_i$ completely. Thus the number of ways that the decoder might FAIL with e $(\geq t+1)$ symbol errors, given $\underline{B}_i = \underline{w}_{i,j}$ and $\underline{\tilde{B}}_i = \underline{w}_{i,j'}$, is $\binom{n-n_t}{e-n_e}$, and we have

$$P(F_i \mid \underline{B}_i = \underline{w}_j, \underline{\tilde{B}}_i = \underline{w}_{j'}) = \sum_{e=t+1}^{n} \binom{n-n_t}{e-n_e} P_{sy}^{e-n_e} (1-P_{sy})^{n-n_t-(e-n_e)}, \tag{3.19}$$

where $P_{sy} = 1-(1-p)^m$ is the symbol error probability of the R-S channel code.

Case 2: Consider the case that $\underline{S}_L$ does not have bit errors in the position that $\underline{B}_i$ occupies but $\underline{S}_R$ has bit errors in the position that $\underline{B}_i$ occupies. Suppose $e\,(\geq t+1)$ symbol errors occurs and the decoder FAIL's when $\underline{B}_i = \underline{w}_{i,j}$ and $\underline{\tilde{B}}_i = \underline{w}_{i,j'}$. In this case, n_e-1 symbol errors are in the symbols that $\underline{B}_i$ occupies completely and 1 symbol error is in $\underline{S}_R$. Thus, $e-n_e$ symbol errors must occur among $\underline{S}_L$ and the symbols that are not completely or partially occupied by $\underline{B}_i$. This implies two further subcases: (1) $e-n_e$ symbol errors occur in the symbols that are not completely or partially occupied by $\underline{B}_i$ and there is no error in $\underline{S}_L$, and (2) $e-n_e$-1 symbol errors occur in the symbols that are not occupied by $\underline{B}_i$ and one symbol error occurs in the m-b_L bits of $\underline{S}_L$ that are not occupied by $\underline{B}_i$. For case (1), there are $\binom{n-n_t}{e-n_e}$ ways for the decoder to FAIL and each has the probability $P_{sy}^{e-n_e}(1-P_{sy})^{n-n_t-(e-n_e)}$. Also, for case (2), we have $\binom{n-n_t}{e-n_e-1}$ ways for the decoder to FAIL and each has the probability $P_{sy}^{e-n_e}(1-P_{sy})^{n-n_t-(e-n_e-1)}$. Thus, we have

$$P(F_i \mid \underline{B}_i = \underline{w}_j, \underline{\tilde{B}}_i = \underline{w}_{j'}) = \sum_{e=t+1}^{n} \left\{ \binom{n-n_t}{e-n_e} P_{sy}^{e-n_e}(1-P_{sy})^{n-n_t-(e-n_e)}(1-P_{sy,L}) \right. \\ \left. + \binom{n-n_t}{e-n_e-1} P_{sy}^{e-n_e-1}(1-P_{sy})^{n-n_t-(e-n_e-1)} P_{sy,L} \right\}, \tag{3.20}$$

where $P_{sy,L}$ is defined in (3.18).

Case 3: The probability $P(F_i \mid \underline{B}_i = \underline{w}_{i,j}, \underline{\tilde{B}}_i = \underline{w}_{i,j'})$ for the case that $\underline{S}_R$ does not have bit errors in the positions occupied by $\underline{B}_i$ but $\underline{S}_L$ has bit errors in the positions occupied by $\underline{B}_i$, has an expression similar to (3.20)

$$P(F_i \mid \underline{B}_i = \underline{w}_j, \underline{\tilde{B}}_i = \underline{w}_{j'}) = \sum_{e=t+1}^{n} \left\{ \binom{n-n_t}{e-n_e} P_{sy}^{e-n_e}(1-P_{sy})^{n-n_t-(e-n_e)}(1-P_{sy,R}) \right. \\ \left. + \binom{n-n_t}{e-n_e-1} P_{sy}^{e-n_e-1}(1-P_{sy})^{n-n_t-(e-n_e-1)} P_{sy,R} \right\}. \tag{3.21}$$

Case 4: Finally, consider the case that both $\underline{S}_L$ and $\underline{S}_R$ do not have bit errors caused by the bit errors between $\underline{B}_i$ and the channel output $\underline{\tilde{B}}_i$. Suppose $e(\geq t+1)$ symbol errors occurs. For this case, we consider four subcases depending on whether the m-b_L and m-b_R bits unoccupied by $\underline{B}_i$ in $\underline{S}_L$ and $\underline{S}_R$, respectively, cause or do not cause symbol errors in $\underline{S}_L$ and $\underline{S}_R$. (a) When there are no bit errors in the m-b_L and m-b_R bits unoccupied by $\underline{B}_i$ in $\underline{S}_L$ and $\underline{S}_R$, respectively, $e-n_e$ symbol errors must occur in the symbols which are not completely or partially occupied by $\underline{B}_i$. Note that the number of such symbols is $n-n_t$. (b) When both $\underline{S}_L$ and $\underline{S}_R$ have bit errors, $e-n_e$-2 symbol errors must occur in the symbols which are not completely or partially occupied by $\underline{B}_i$. (c)(d) When either $\underline{S}_L$ or $\underline{S}_R$ has bit errors, (i.e., there are no bit errors in the m-b_L $(m-b_R)$ bits which are not occupied by $\underline{B}_i$ in $\underline{S}_L$ $(\underline{S}_R)$ and there are bit errors in the m-b_R $(m-b_L)$ bits which are not occupied by $\underline{B}_i$ in $\underline{S}_R$ $(\underline{S}_L)$), then $e-n_e$-1 symbol errors must occur in the symbols which are not occupied by $\underline{B}_i$, since there is a symbol error in $\underline{S}_R$ $(\underline{S}_L)$. Thus, the probability $P(F_i \mid \underline{B}_i = \underline{w}_{i,j}, \underline{\tilde{B}}_i = \underline{w}_{i,j'})$ is expressed by the sum of four terms:

$$
\begin{aligned}
&P(F_i \mid \underline{B}_i = \underline{w}_{i,j}, \underline{\tilde{B}}_i = \underline{w}_{i,j'}) \\
&\quad = \sum_{e=t+1}^{n} \Bigg\{ \binom{n-n_t}{e-n_e} P_{sy}^{e-n_e} (1-P_{sy})^{n-n_t-(e-n_e)} (1-P_{sy,L})(1-P_{sy,R}) \\
&\qquad + \binom{n-n_t}{e-n_e-1} P_{sy}^{e-n_e-1} (1-P_{sy})^{n-n_t-(e-n_e-1)} P_{sy,L} (1-P_{sy,R}) \\
&\qquad + \binom{n-n_t}{e-n_e-1} P_{sy}^{e-n_e-1} (1-P_{sy})^{n-n_t-(e-n_e-1)} (1-P_{sy,L}) P_{sy,R} \\
&\qquad + \binom{n-n_t}{e-n_e-2} P_{sy}^{e-n_e-2} (1-P_{sy})^{n-n_t-(e-n_e-2)} P_{sy,L} P_{sy,R} \Bigg\}.
\end{aligned}
\tag{3.22}
$$

We now combine the expressions for the above four cases, which are given by (3.19) - (3.22). Let (3.19) - (3.22) be $P_{F_i,11}$, $P_{F_i,01}$, $P_{F_i,10}$, and $P_{F_i,00}$, respectively, and define a function $I(\underline{S})$ as

$$I(\underline{S}) = \begin{cases} 1, \text{ if symbol } \underline{S} \text{ has any bit error in the bits occupied by } \underline{B}_i \\ 0, \text{ else.} \end{cases} \tag{3.23}$$

Then,

$$\begin{aligned} P(F_i | \underline{B}_i = \underline{w}_{i,j}, \underline{\tilde{B}}_i = \underline{w}_{i,j'}) &= I(\underline{S}_L) I(\underline{S}_R) P_{DF,11} + \left(1 - I(\underline{S}_L)\right) I(\underline{S}_R) P_{DF,01} \\ &\quad + I(\underline{S}_L)\left(1 - I(\underline{S}_R)\right) P_{DF,10} + \left(1 - I(\underline{S}_L)\right)\left(1 - I(\underline{S}_R)\right) P_{DF,00}. \end{aligned} \tag{3.24}$$

With some manipulation, we get a final expression for $P(F_i | \underline{B}_i = \underline{w}_{i,j}, \underline{\tilde{B}}_i = \underline{w}_{i,j'})$:

$$\begin{aligned} &P(F_i | \underline{B}_i = w_{i,j}, \underline{\tilde{B}}_i = w_{i,j'}) = \\ &\sum_{e=t+1}^{n} \Bigg\{ \binom{n-n_t}{e-n_e} P_{sy}^{e-n_e} (1-P_{sy})^{n-n_e-(e-n_e)} (1-P_{sy,L}^I)(1-P_{sy,R}^I) \\ &+ \binom{n-n_t}{e-n_e-1} P_{sy}^{e-n_e-1} (1-P_{sy})^{n-n_e-(e-n_e-1)} \left((1-P_{sy,L}^I) P_{sy,R}^I + P_{sy,L}^I (1-P_{sy,R}^I)\right), \\ &+ \binom{n-n_t}{e-n_e-2} P_{sy}^{e-n_e-2} (1-P_{sy})^{n-n_e-(e-n_e-2)} P_{sy,L}^I P_{sy,R}^I \Bigg\} \end{aligned} \tag{3.25}$$

where $P_{sy,L}^I = P_{sy,L}$ if there are no errors in the bits of $\underline{S}_L$ occupied by $\underline{B}_i$, otherwise, $P_{sy,L}^I = 0$. And $P_{sy,R}^I$ is defined similarly.

B. The case that $\underline{B}_i$ is spread over two R-S codewords

In the case that $\underline{B}_i$ is spread over two R-S codewords as shown in Figure 3.4, the event F_i is the union of three events: $(NF_{i,1}, F_{i,2})$, $(F_{i,1}, NF_{i,2})$, and $(F_{i,1}, F_{i,2})$, where $(NF_{i,1}, F_{i,2})$ is the event that the first received sequence FAILS and the second does not, and the other two events are defined similarly. Thus, $P_i(\underline{w}_{i,j'} | \underline{w}_{i,j}, F_i)$ becomes

$$\begin{aligned} P_i(\underline{w}_{i,j'} | \underline{w}_{i,j}, F_i) &= \frac{P(\underline{w}_{i,j'}, \underline{w}_{i,j}, F_i)}{P(\underline{w}_{i,j}, F_i)} = \frac{P(\underline{w}_{i,j'}, \underline{w}_{i,j}, F_i)}{P(\underline{w}_{i,j}) P(F_i)} \\ &= \frac{P(\underline{w}_{i,j}, \underline{w}_{i,j'}, F_{i,1}, NF_{i,2}) + P(\underline{w}_{i,j}, \underline{w}_{i,j'}, NF_{i,1}, F_{i,2}) + P(\underline{w}_{i,j}, \underline{w}_{i,j'}, F_{i,1}, F_{i,2})}{P(\underline{w}_{i,j}) P(F_i)}. \end{aligned} \tag{3.26}$$

In the above equation, $P(\underline{w}_{i,j}, \underline{w}_{i,j'}, F_{i,1}, NF_{i,2})$ becomes

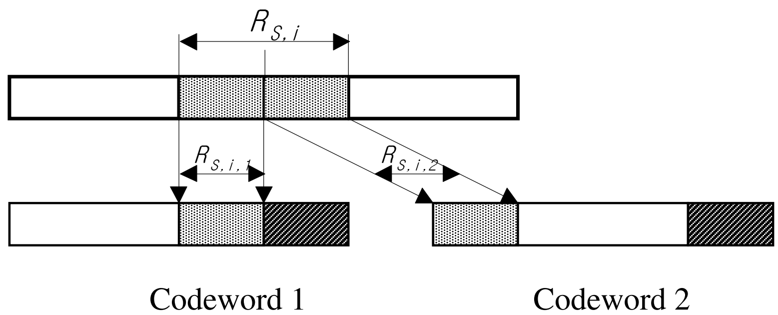


Figure 3.4: The case that $\underline{B}_i$ is spread over two R-S codewords

$$
\begin{aligned}
P(\underline{w}_{i,j},\underline{w}_{i,j'},F_{i,1},NF_{i,2}) &= P(\underline{w}_{i,j})P(F_{i,1},NF_{i,2} \mid \underline{w}_{i,j})P(\underline{w}_{i,j'} \mid \underline{w}_{i,j},F_{i,1},NF_{i,2}) \\
&= P(\underline{w}_{i,j})P(F_{i,1})P(NF_{i,2})P(\underline{w}_{i,j'} \mid \underline{w}_{i,j},F_{i,1},NF_{i,2}),
\end{aligned}
\tag{3.27}
$$

and the other two terms, $P(\underline{w}_{i,j},\underline{w}_{i,j'},F_{i,1},NF_{i,2})$ and $P(\underline{w}_{i,j},\underline{w}_{i,j'},F_{i,1},F_{i,2})$, have similar expressions. Thus, we have

$$
\begin{aligned}
P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_i) = \frac{1}{P(F_i)}\Big\{ & P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_{i,1},NF_{i,2})P(F_{i,1},NF_{i,2}) \\
& + P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},NF_{i,1},F_{i,2})P(NF_{i,1},F_{i,2}) + P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_{i,1},F_{i,2})P(F_{i,1},F_{i,2}) \Big\}.
\end{aligned}
\tag{3.28}
$$

We now derive the expression for $P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_{i,1},NF_{i,2})$, $P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},NF_{i,1},F_{i,2})$ and $P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_{i,1},F_{i,2})$. Let $R_{S,i,1}$ ($R_{S,i,2}$) denote the number of bits of $\underline{B}_i$ contained in the first (second) codeword, let $\underline{B}_{i,1}$ ($\underline{B}_{i,2}$) denote the binary sequence represented by $R_{S,i,1}$ ($R_{S,i,2}$) bits in the first (second) codeword, and let $\underline{w}_{i,j,1}$ ($\underline{w}_{i,j,2}$) denote $R_{S,i,1}$-bit ($R_{S,i,2}$-bit) binary sequence representing the corresponding integer value among $\{0, 1, \cdots, 2^{R_{S,i,1}}-1\}$ ($\{0, 1, \cdots, 2^{R_{S,i,2}}-1\}$) when an $R_{S,i}$-bit binary sequence $\underline{w}_{i,j}$ representing integer value j-1 is split into $R_{S,i,1}$ bits and $R_{S,i,2}$ bits. Consider the event that the first codeword FAILS and the second does not FAIL. Since the bit errors in first and second codewords are independent, we have

$$
\begin{aligned}
&P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_{i,1},NF_{i,2})P_i(F_{i,1},NF_{i,2}) \\
&= P_i(\underline{\hat{B}}_{i,1} = \underline{w}_{i,j',1} \mid \underline{B}_{i,1} = \underline{w}_{i,j,1}, F_{i,1}) \bullet \\
&\qquad P_i(\underline{\hat{B}}_{i,2} = \underline{w}_{i,j',2} \mid \underline{B}_{i,2} = \underline{w}_{i,j,2}, NF_{i,2}) P_i(F_{i,1}) P_i(NF_{i,2}) \qquad (3.30)\\
&= P_i(\underline{\tilde{B}}_{i,1} = \underline{w}_{i,j',1} \mid \underline{B}_{i,1} = \underline{w}_{i,j,1}, F_{i,1}) \bullet \\
&\qquad P_i(\underline{\hat{B}}_{i,2} = \underline{w}_{i,j',2} \mid \underline{B}_{i,2} = \underline{w}_{i,j,2}, NF_{i,2}) P_i(F_{i,1}) P_i(NF_{i,2}),
\end{aligned}
$$

where the last equality comes from the fact, $\underline{\hat{B}}_{i,1} = \underline{\tilde{B}}_{i,1}$, based on our decoding rule when the codewords FAIL. Since the second codeword does not FAIL, we have $\underline{\hat{B}}_{i,2} = \underline{B}_{i,2}$. Thus, (3.30) becomes

$$
\begin{aligned}
&P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_{i,1},NF_{i,2})P(F_{i,1},NF_{i,2}) \qquad (3.31)\\
&= \begin{cases} P_i(\underline{\tilde{B}}_{i,1} = \underline{w}_{i,j',1} \mid \underline{B}_{i,1} = \underline{w}_{i,j,1}, F_{i,1}) P(F_{i,1}) P(NF_{i,2}), \text{if } \underline{w}_{i,j',2} = \underline{w}_{i,j,2} \\ 0, \qquad \text{otherwise.} \end{cases}
\end{aligned}
$$

Similarly, for the event $(NF_{i,1}, F_{i,2})$, we have

$$
\begin{aligned}
&P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},NF_{i,1},F_{i,2})P(NF_{i,1},F_{i,2}) \qquad (3.32)\\
&= \begin{cases} P_i(\underline{\tilde{B}}_{i,2} = \underline{w}_{i,j',2} \mid \underline{B}_{i,2} = \underline{w}_{i,j,2}, F_{i,2}) P(NF_{i,1}) P(F_{i,2}), \text{if } \underline{w}_{i,j',1} = \underline{w}_{i,j,1} \\ 0, \qquad \text{otherwise,} \end{cases}
\end{aligned}
$$

and for the event $(F_{i,1}, F_{i,2})$, we have

$$
\begin{aligned}
&P_i(\underline{w}_{i,j'}|\underline{w}_{i,j},F_{i,1},F_{i,2})P(F_{i,1},F_{i,2}) \\
&\qquad = P_i(\underline{\tilde{B}}_{i,1} = \underline{w}_{i,j',1} \mid \underline{B}_{i,1} = \underline{w}_{i,j,1}, F_{i,1}) \bullet \qquad (3.33)\\
&\qquad\qquad P_i(\underline{\tilde{B}}_{i,2} = \underline{w}_{i,j',2} \mid \underline{B}_{i,2} = \underline{w}_{i,j,2}, F_{i,2}) P(F_{i,1}) P(F_{i,2}).
\end{aligned}
$$

Inserting (3.31) – (3.33) into (3.28), we obtain an expression for $P_i(\underline{w}_{i,j'} \mid \underline{w}_{i,j,} F_i)$ in (3.34). The expressions for $P_i(\underline{\tilde{B}}_{i,1} = \underline{w}_{i,j',1} \mid \underline{B}_{i,1} = \underline{w}_{i,j,1}, F_{i,1})$ and

$P_i(\underline{\tilde{B}}_{i,2} = \underline{w}_{i,j',2} \mid \underline{B}_{i,2} = \underline{w}_{i,j,2}, F_{i,2})$ are obtained using analysis similar to that used to find $P_i(\underline{w}_{i,j'} | \underline{w}_{i,j}, F_i)$ in the case that $\underline{B}_i$ is entirely contained within one R-S codeword.

$$P_i(\underline{w}_{i,j'} | \underline{w}_{i,j}, F_i) =$$

$$\begin{cases} \frac{1}{P(F_i)} P_i(\underline{\tilde{B}}_{i,1} = \underline{w}_{i,j',1} \mid \underline{B}_{i,1} = \underline{w}_{i,j,1}, F_{i,1}) P(F_{i,1}) \bullet \\ \quad P_i(\underline{\tilde{B}}_{i,2} = \underline{w}_{i,j',2} \mid \underline{B}_{i,2} = \underline{w}_{i,j,2}, F_{i,2}) P(F_{i,2}), & \text{if } \underline{w}_{i,j',1} \neq \underline{w}_{i,j,1}, \underline{w}_{i,j',2} \neq \underline{w}_{i,j,2} \\ \frac{1}{P(F_i)} P_i(\underline{\tilde{B}}_{i,1} = \underline{w}_{i,j',1} \mid \underline{B}_{i,1} = \underline{w}_{i,j,1}, F_{i,1}) P(F_{i,1}) \bullet \\ \quad \left\{ P(NF_{i,2}) + P_i(\underline{\tilde{B}}_{i,2} = \underline{w}_{i,j',2} \mid \underline{B}_{i,2} = \underline{w}_{i,j,2}, F_{i,2}) P(F_{i,2}) \right\}, & \text{if } \underline{w}_{i,j',1} \neq \underline{w}_{i,j,1}, \underline{w}_{i,j',2} = \underline{w}_{i,j,2} \\ \frac{1}{P(F_i)} P_i(\underline{\tilde{B}}_{i,2} = \underline{w}_{i,j',2} \mid \underline{B}_{i,2} = \underline{w}_{i,j,2}, F_{i,2}) P(F_{i,2}) \\ \quad \left\{ P(NF_{i,1}) + P_i(\underline{\tilde{B}}_{i,1} = \underline{w}_{i,j',1} \mid \underline{B}_{i,1} = \underline{w}_{i,j,1}, F_{i,1}) P(F_{i,1}) \right\} & \text{if } \underline{w}_{i,j',1} = \underline{w}_{i,j,1}, \underline{w}_{i,j',2} \neq \underline{w}_{i,j,2} \\ \frac{1}{P(F_i)} \left\{ P_i(\underline{\tilde{B}}_{i,1} = \underline{w}_{i,j',1} \mid \underline{B}_{i,1} = \underline{w}_{i,j,1}, F_{i,1}) P(F_{i,1}) P_i(\underline{\tilde{B}}_{i,2} = \underline{w}_{i,j',2} \mid \underline{B}_{i,2} = \underline{w}_{i,j,2}, F_{i,2}) P(F_{i,2}) \right. \\ \quad + P_i(\underline{\tilde{B}}_{i,1} = \underline{w}_{i,j',1} \mid \underline{B}_{i,1} = \underline{w}_{i,j,1}, F_{i,1}) P(F_{i,1}) P(NF_{i,2}) \\ \quad \left. + P_i(\underline{\tilde{B}}_{i,2} = \underline{w}_{i,j',2} \mid \underline{B}_{i,2} = \underline{w}_{i,j,2}, F_{i,2}) P(NF_{i,1}) P(F_{i,2}) \right\} & \text{if } \underline{w}_{i,j',1} = \underline{w}_{i,j,1}, \underline{w}_{i,j',2} = \underline{w}_{i,j,2} \end{cases} \tag{3.34}$$

3.4 Complexity

In this section, we consider the complexity of the tandem source-channel coding system and channel-optimized transform coding system. As a measure of complexity we sum the number of arithmetic operations needed per sample to perform encoding and decoding. For simplicity we do not distinguish multiplications, additions and comparisons. We do not expect that doing so would change the qualitative nature of the results, because the number of operations tends to be proportional to the number of multiplications. Measuring the complexity of the system, the complexity of encoding and decoding of the source and channel coder are added. The rationale for adding encoding and decoding complexity is that even if one is only interested in encoding (decoding), somewhere else, power is being consumed by a decoder (encoder). This is a reasonable model for peer to peer communications. Also, we do not consider the amount of auxiliary storage required for encoding and decoding codes because the systems investigated here do not use much.

3.4.1 Channel-Optimized Transform Code

The complexity of the channel-optimized transform code is the sum of the complexities for the transform and the quantizers.

The transform : The matrix multiplication in transform encoding and decoding requires L^2 multiplications and $L(L-1)$ additions for processing each block of L data samples. Summing these for both the encoder and decoder, and dividing by the number of source sample L, we have 4L-2 operations per sample.

The quantizers: Given a value of V_i the closest of $2^{R_{S,i}}$ quantization levels can be found via a binary search (i.e. tree-structured quantization) with $R_{S,i}$ comparisons to prespecified thresholds. It follows that quantization requires $\frac{1}{L}\sum_{i=1}^{L} R_{S,i} = R_S$ operations per sample. The decoder requires no arithmetic operation.

Summing the complexity of the transform and the quantizers, the complexity is given by

$$C_j = 4L - 2 + R_S \,. \tag{3.35}$$

3.4.2 Channel-Optimized Transform Code with Discrete Cosine Transform

We have so far considered the conventional KL transform code as the source coder in tandem source-channel coding system. However, the discrete cosine transform (DCT) is widely used in many applications, it is desirable to investigate the performance of a tandem system consisting of a DCT based source coder and an RS code. Even though the DCT does not provide optimal performance, as does the KL transform code, it generally provides good performance and above all, low computational complexity.

One of the well-known fast computational algorithms [29], [30] requires $(L/2)\log_2 L$ multiplications and $(3L/2)\log_2 L - L + 1$ additions for transform size *N*. Since the computational complexity of the DCT is smaller than that of KLT, under a given complexity constraint we can allocate more complexity into the channel code or we can increase *L* and we might obtain improved performance. Summing these complexity for both the encoder and decoder, and dividing by the number of source samples *L*, we obtain

$$4\log_2 L\text{-}2\text{+}2/L \text{ operations per sample.} \tag{3.36}$$

Thus, the channel-optimized transform code with the DCT can provide a lot of complexity saving. The complexity is given by

$$C_{j,DCT} = 4\log_2 L - 2 + 2/L + R_S\,. \tag{3.37}$$

3.4.3 Tandem source-channel code

The complexity of the tandem source-channel coding is the sum of the complexities for the transform, the quantizers and the RS code. The complexities of the transform and the quantizers are given in the previous section. RS encoding and decoding require discrete polynomial operations unlike the floating point operations of the source code. Nevertheless, we count them in the same way. RS encoding performs the polynomial multiplication of an (n-k)th-degree generator polynomial and a kth-degree message polynomial. We need about $2k(n-k)$ operations per codeword for R-S encoding.

RS decoding is performed in 4 steps: (1) syndrome calculation, (2) finding the error-locator polynomial, (3) finding the error locations, and (4) finding the error values. Table 3.1 summarizes the required operations for each step of RS decoding [18]. For these calculations we assume that the Berlekamp-Messey shift register synthesis decoding algorithm is used to find the error-locator polynomial; the Chien search algorithm is used to find the roots of the error-locator polynomial; and the error values are evaluated using Forney's algorithm. In total, the complexity for R-S decoding is approximately

$$3(n-k)n+5.5(n-k)^2 \text{ operations per codeword.} \tag{3.38}$$

Since km/R_S data samples are encoded on the average by one R-S codeword, the number of operations per sample for R-S encoding and decoding is

$$(2k+3n+5.5(n-k))(n-k)R_S/km. \tag{3.39}$$

Summing the complexity of the transform, the quantizers and the RS code, we find the number of operations per data sample for the tandem system as

$$C = C(L, R_S, n, k, m) = 4L - 2 + R_S + \frac{(2k+3n+5.5(n-k))(n-k)R_S}{km} \tag{3.40}$$

3.4.4 Tandem source-channel code with Discrete Cosine Transform

If a DCT is used instead of the KLT, the complexity for the transform need to be replaced with (3.36). Then, the number of operations per data sample for the tandem system with the DCT is

$$C_{t,DCT} = 4\log_2 L - 2 + 2/L + R_S + \frac{(2k+3n+5.5(n-k))(n-k)R_S}{km}. \tag{3.41}$$

TABLE 3.1
The number of operations for R-S decoding

Step	Multiplications	Additions
1	2tn	2t(n-1)
2	$8t^2$	$8t^2$
3	tn	t(n-1)
4	$3t^2$	$3t^2$
Total	$3tn+11t^2$ $= 3/2\ (n-k)n + 11/4\ (n-k)^2$	$3t(n-1)+11t^2$ $= 3/2\ (n-k)(n-1) + 11/4\ (n-k)^2$

3.5 Optimization Procedure

3.5.1 Channel-Optimized Transform Code

The purpose of optimization in the channel-optimized transform coding is to find the transform parameter L that gives the minimum distortion under the complexity constraint. Since the complexity of channel-optimized transform code is directly proportional to the transform dimension as shown in (3.35) and the performance of the system improves with the transform dimension, the optimization procedure need only find the largest transform dimension satisfying the complexity constraint and compute the distortion of the transform code with that transform dimension. Thus the optimization procedure is summarized as follows. For specific choices of the source parameter ρ and channel parameters R and p, and a complexity constraint c, a computer program evaluates the distortion of the transform code for the largest transform dimension L in the set $\{1,2,4,8,\ldots,128\}$ such that the complexity given by (3.35) is no larger than c.

3.5.2 Tandem source-channel coding

The tandem source-channel coding is optimized as follow. For specific values of the source parameter ρ and channel parameters R and p, and a complexity constraint c, a computer program searches for the best choices of source code parameters L, R_S and channel code parameters n, k and m subject to the constraints $nR_S / R \leq k$, LR_S is an integer, $n = 2^m - 1$, $1 \leq k \leq n$ and $C(L,R_S,n,k,m) \leq c$. That is, we optimize our system using an exhaustive search algorithm. To do so, the program uses the expressions of Section 3.3 and 3.4 to compute the MSE and complexity with parameter values varying in three nested loops. The values of L come from the set $\{1,2,4,8,16,...,128\}$; the values of R_S are such that LR_S equals $1,2,\cdots,\lfloor RL \rfloor$; the values of m come from the set $\{1,2,...,12\}$; and finally, $k = \lceil nR_S / R \rceil$. Note that for given choices of L, R_S, k and m, the program must choose M_S and M_C as described in Section 3.3.

3.6 Results

Figure 3.5 shows the results of the optimization for the case that the source has correlation coefficient $\rho = 0.9$ and the BSC channel rate is 5 bits/source sample. There are five sets of curves. The dash-dotted curves show the signal-to-noise ratio for the transform codes with rate $R_S = R = 5$ bits/sample, no channel coding, transform code dimensions L = 1, 32, 64 and 128 and complexities c = 5, 131,259 and 515, respectively. It can be seen that using large transform dimension has substantial benefits when the channel is quite reliable (e.g. $p \leq 10^{-5}$), but the benefits decrease as the channel becomes noisier, to the point where there is little reason to choose anything but scalar quantization when $p \geq 10^{-3}$.

The bold dash-dotted line shows the performance of a tandem source-channel code consisting of the conventional KL transform code with L = 128 and an ideal channel code (the transform code has rate five times the capacity of the BSC and its encoded bits are assumed to be unaffected by channel coders), and the bold dotted line shows the best possible performance of any tandem source-channel code (the Shannon's rate-distortion function of the source evaluated at five times the capacity of the BSC) as mentioned in Section 2.3. Note that there is about 4.34 dB performance gap between two curves.

The solid curves of Figure 3.5 show the results of the optimization for the tandem source-channel code for c = 35, 67, 131, 259 and 515. We can see from these curves that performance improves substantially with increasing complexity. For small values of p, little channel coding is needed, so the complexity required to do well is mainly that of the source code whose dimension should be chosen large. As p increases, there is more benefit to increasing the complexity, principally that of the channel code, and the source coding complexity decreases, principally through decreasing L, but secondarily by decreasing R_S. Other details for tandem source-channel code are provided in Figure 3.6, which for systems optimized with various complexity constraints, shows the fraction of complexity c_S / c that is invested in the source code, the channel coding rate R_C, the channel code blocklength n, the channel symbol size m, and the transform dimension L.

The dotted lines in Figure 3.5 show the performance of the channel-optimized transform code for c = 5, 19 and 515, which correspond to L = 1, 4, 128. As with the tandem source-channel code, we can see that performance improves substantially with increasing complexity at all p's. As p increases, there is more benefit to increasing the complexity and the benefit stays roughly the same with p.

For BSC error probability p=10^{-2} and several values of total complexity c, Figure 3.7 shows how the SNR of tandem coding varies with the proportion of complexity invested in the source code (by varying L). We see from this that the highest SNR is obtained with about 15-30% of the complexity invested in the source code. We also note the rather surprising flatness of the curves for higher complexity, indicating that good performance at a given complexity can be obtained with a variety of system parameters.

A representative result of performance comparison between the tandem source-channel code and the channel-optimized transform code under complexity constraints is provided in Figure 3.8. This plots the performance of channel-optimized transform coding and of tandem coding for the case that the source has correlation coefficient ρ = 0.9, the BSC has error probability p = 10^{-2}, and the channel is used R = 5 times per data sample. The bottom line shows the performance of transform coding optimized without regard to the channel, but used directly on the channel without channel coding. It is easy to see that there is no benefit to increasing the complexity, i.e. the transform dimension. However, as complexity increases, the performance of channel-optimized transform coding improves. Specifically, there is about a 4 dB gain over conventional transform coding for $c \cong 10$, and the gain increases until it saturates at about 11 dB for $c \cong$ 50. At this point, there is no benefit to further increasing the transform dimension.

Not surprisingly, the performance of optimized tandem codes also increases with complexity. For small values the performance is not as good as that of channel-optimized transform coding, while for larger values it is better. In particular, tandem coding becomes better at around 80 operations per sample. At this point, although the SNR of channel-optimized transform coding has saturated, dedicating additional arithmetic operations to the channel code permits the tandem code to continue to improve. Thus, the results suggest there is a complexity threshold such that when the number of available operations per data sample is above this threshold, tandem coding is better, and when below the threshold, channel-optimized transform coding is better. Also shown in the figure is the performance of a tandem system consisting of conventional transform code with L = 128 and an ideal channel code (the transform code has rate five times the capacity of the BSC and its encoded bits are assumed to be unaffected by channel errors), and also the best possible performance of any tandem code (the Shannon distortion-rate function of the source evaluated at five times the capacity of the BSC).

We repeated the optimization for a variety of other choices for R, p and ρ. Figure 3.9 (a), (b) and (c) show the performance comparison between channel-optimized transform coding and tandem source-channel coding for R = 5, ρ = 0.9, and p = 0.02, 0.01 and 0.001, respectively. As shown in these figures, complexity thresholds become smaller as p decreases. This trend is also observed for other values of R. Tables 3.2 summarizes this trend by showing the thresholds and the gain of the tandem system over the joint system at 2 times the threshold. The latter is intended as an indicator of how much can be gained with the additional complexity of tandem coding. We can see that the complexity thresholds decrease and the gains increase as the channel becomes more reliable. Figure 3.10 shows the change of complexity thresholds as R varies. It plots the performance of the joint and tandem systems for ρ = 0.9, p = 0.01, and R = 3, 4 and 5. We notice that the complexity thresholds decrease and the gains increase as the channel becomes more reliable and more channel uses are available. This is because the channel coding can cheaply improve the SNR. The detailed gains are shown in Tables 3.2. This trend on the thresholds and gains means that contrary to what one might initially think, channel coding is more immediately useful, i.e. useful with moderate complexity, when the channel is more reliable.

Figure 3.11 shows the variation of delay thresholds as ρ varies by plotting the performance of joint and tandem systems for p = 0.01, R = 5, and ρ = 0.70, 0.90 and 0.95. Table 3.3 shows the complexity thresholds and the gain of the tandem system over the joint system at 2 times the threshold for R = 5, various p's and various ρ 's. The gains increase as the source becomes more correlated.

Figure 3.12 shows the performance of joint and tandem source-channel code with the Discrete Cosine Transform. We can see that the performance of the system with the DCT under a complexity constraints is better than that of the system with the KLT. This improvement comes from low computational complexity of tDCT as discussed in Section 3.4.2. The low complexity outweighs supoptimality of DCT and allows large L. The performance improvement of DCT over KLT becomes higher as the complexity increases.

In summary, the distortion vs. complexity performance of representative tandem and joint source-channel codes has been computed and compared. For small values of complexity, channel-optimized transform coding is better. However, as complexity increases, its performance tends to saturate and tandem coding becomes better. In other words there exists a complexity threshold such that tandem coding is better than channel-optimized transform coding when the available complexity is above this threshold. Conversely, channel-optimized transform coding is better when the available complexity is under this threshold.

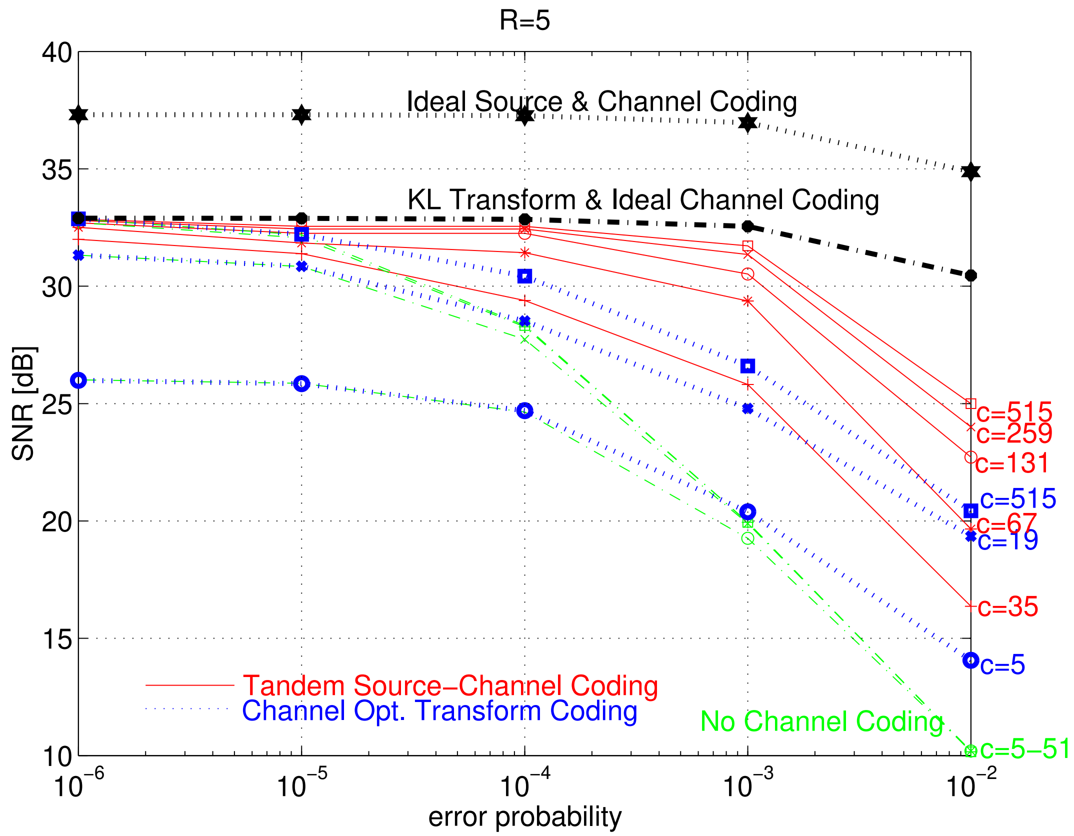


Figure 3.5: For G-M source with $\rho = 0.9$ and BSC with rate R=5, SNR [dB] is plotted vs. BSC error probability p for transform coding with no channel coding (dashed lines), complexity optimized tandem coding (solid lines), ideal channel coding with L=128 transform coding (dotted line), and ideal source and channel coding (dotted line).

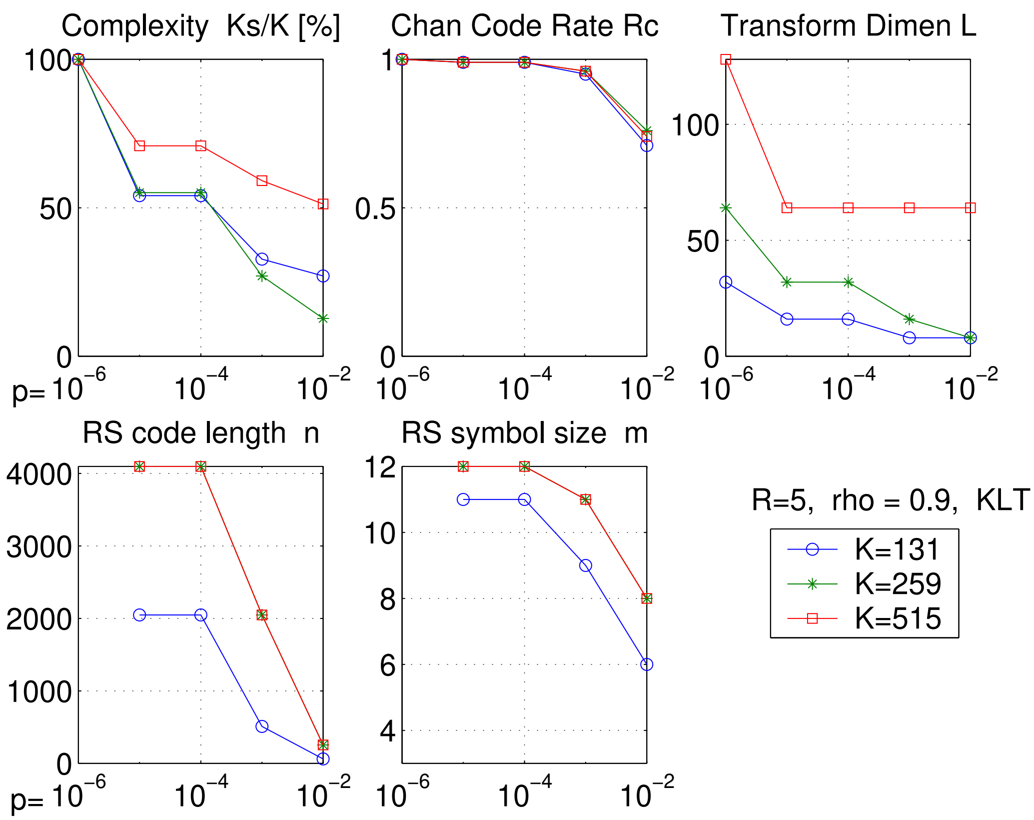


Figure 3.6: Details of the systems whose performance are shown in Figure 3.5

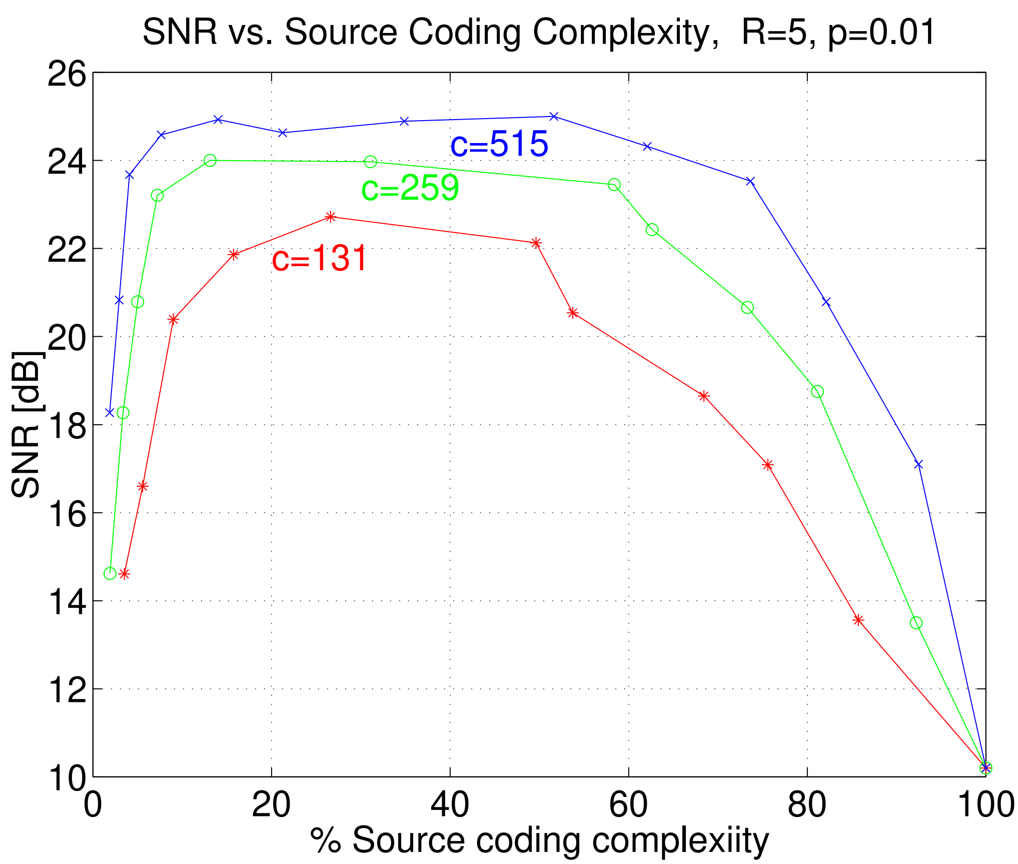


Figure 3.7: SNR vs. percentage of complexity allocated to the source code, for several values of overall complexity, $\rho = 0.9$ and p=10^{-2}

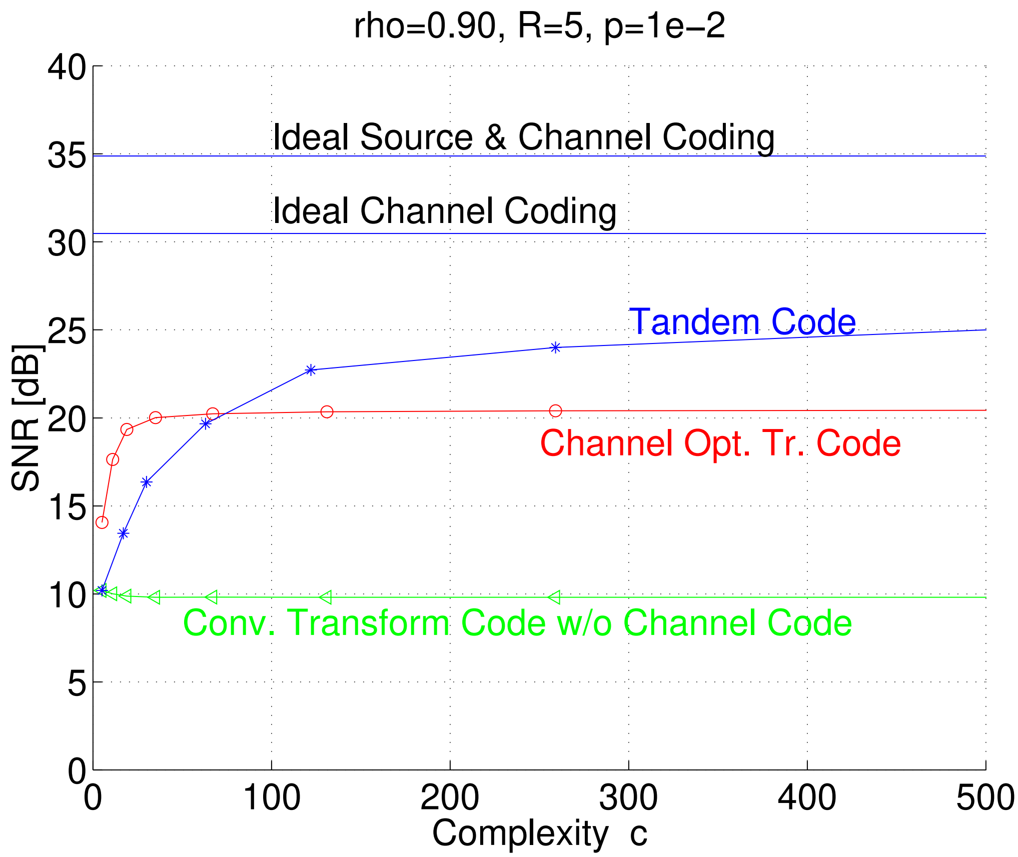
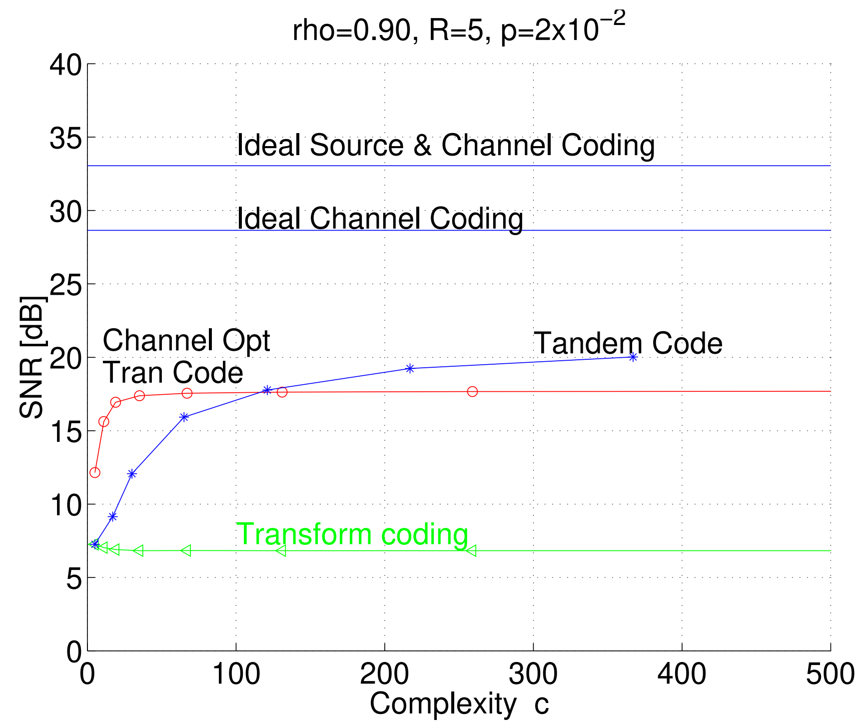
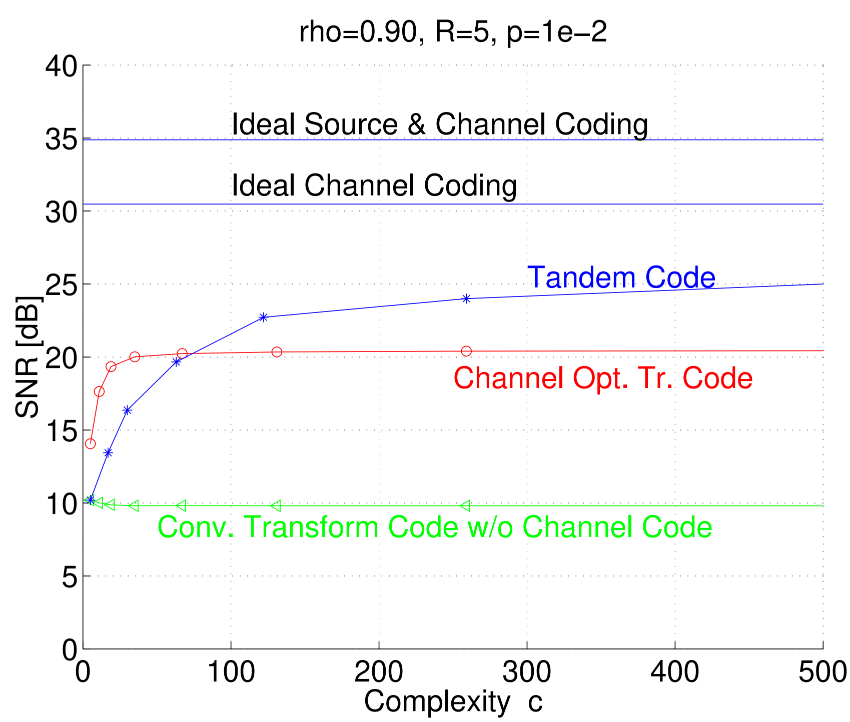


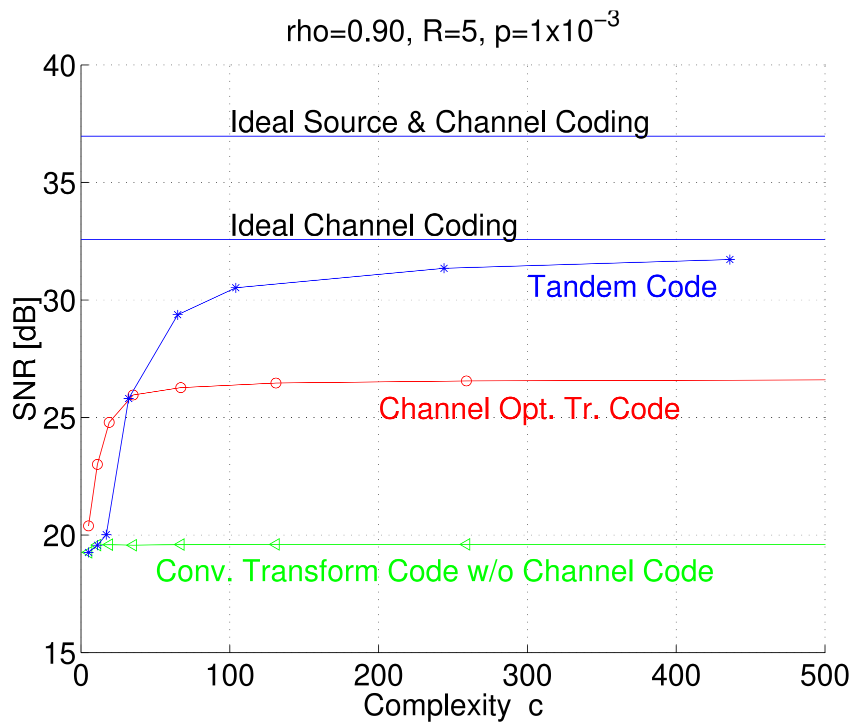
Figure 3.8: The performance of channel-optimized transform coding and tandem coding for R=5, p=10^{-2} and ρ =0.9


rho=0.90, R=5, p=2x10^−2
Ideal Source & Channel Coding
Ideal Channel Coding
Channel Opt
Tran Code
Tandem Code
Transform coding
SNR [dB]
Complexity c
0
5
10
15
20
25
30
35
40
0
100
200
300
400
500

(a)

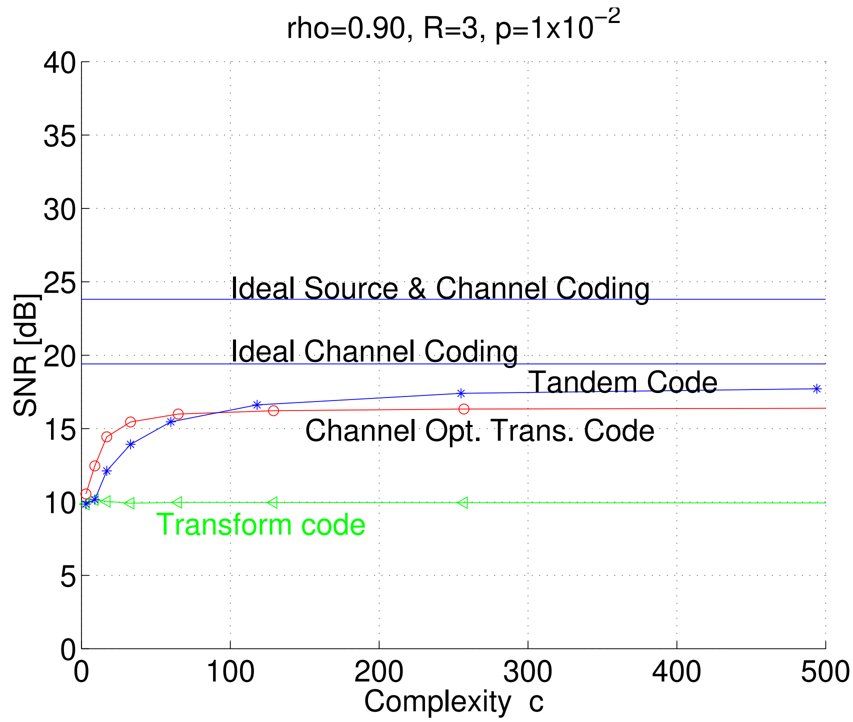

rho=0.90, R=5, p=1e−2
Ideal Source & Channel Coding
Ideal Channel Coding
Tandem Code
Channel Opt. Tr. Code
Conv. Transform Code w/o Channel Code
SNR [dB]
Complexity c
0
5
10
15
20
25
30
35
40
0
100
200
300
400
500

(b)

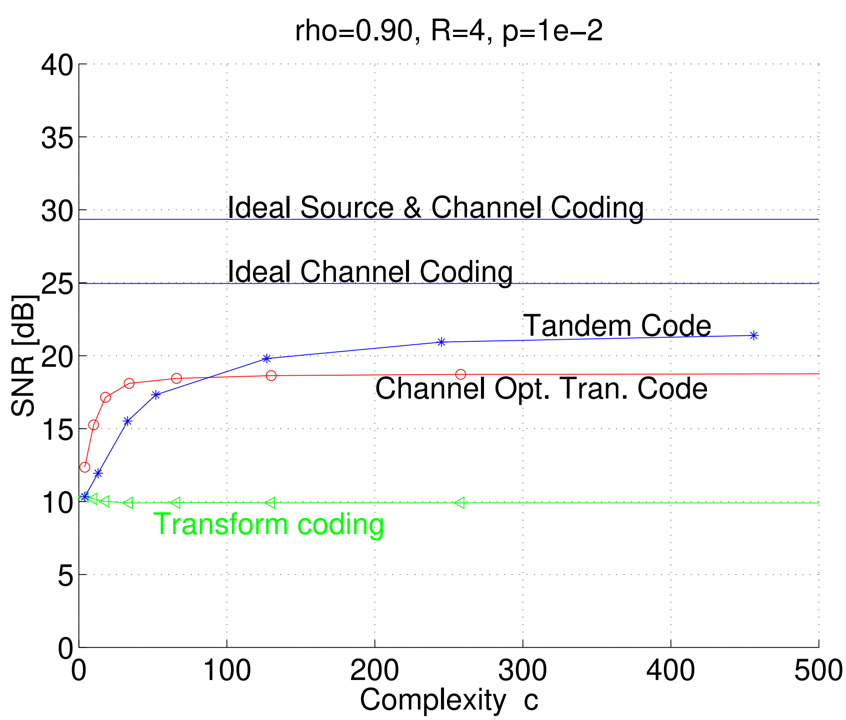


(c)

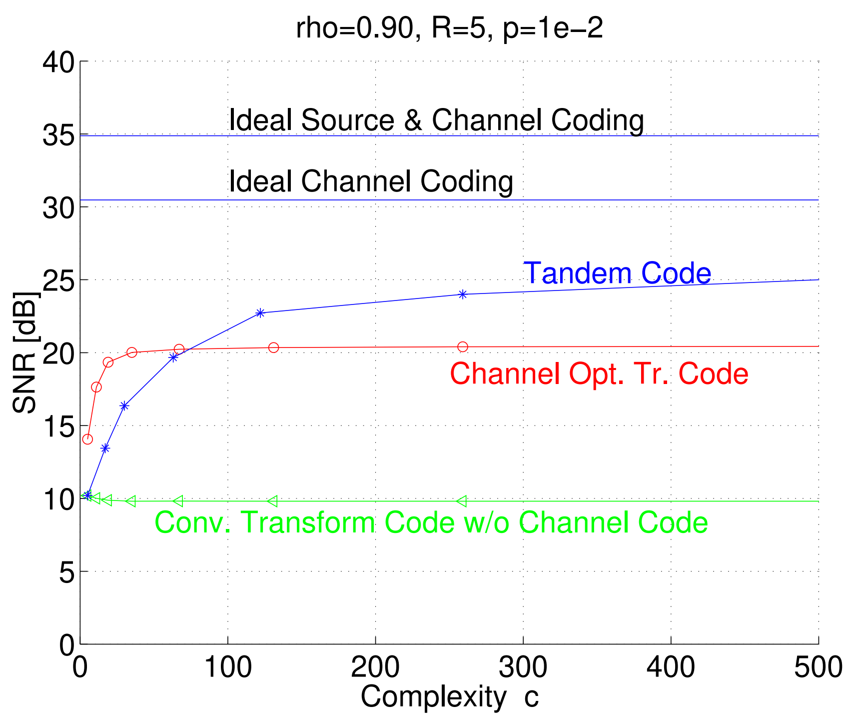
Figure 3.9: A comparison between channel-optimized transform coding and tandem source-channel coding for R=5 and $\rho = 0.9$. (a) p=0.02 (b) p=0.01 (c) p=0.001


rho=0.90, R=3, p=1x10^-2
Ideal Source & Channel Coding
Ideal Channel Coding
Tandem Code
Channel Opt. Trans. Code
Transform code
SNR [dB]
Complexity c
0
5
10
15
20
25
30
35
40
0
100
200
300
400
500

(a)

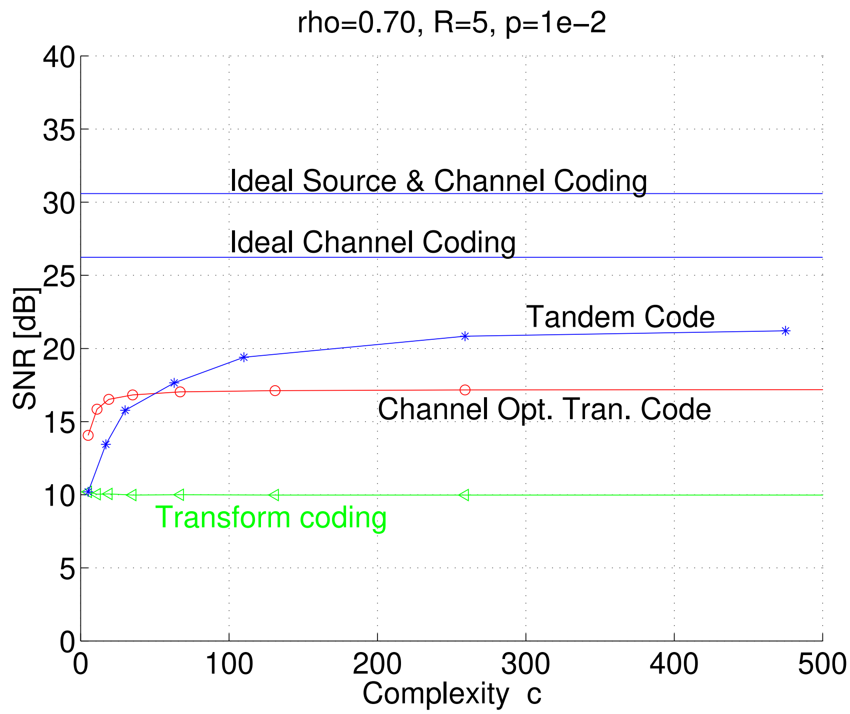

rho=0.90, R=4, p=1e−2
Ideal Source & Channel Coding
Ideal Channel Coding
Tandem Code
Channel Opt. Tran. Code
Transform coding
SNR [dB]
Complexity c
0
5
10
15
20
25
30
35
40
0
100
200
300
400
500

(b)

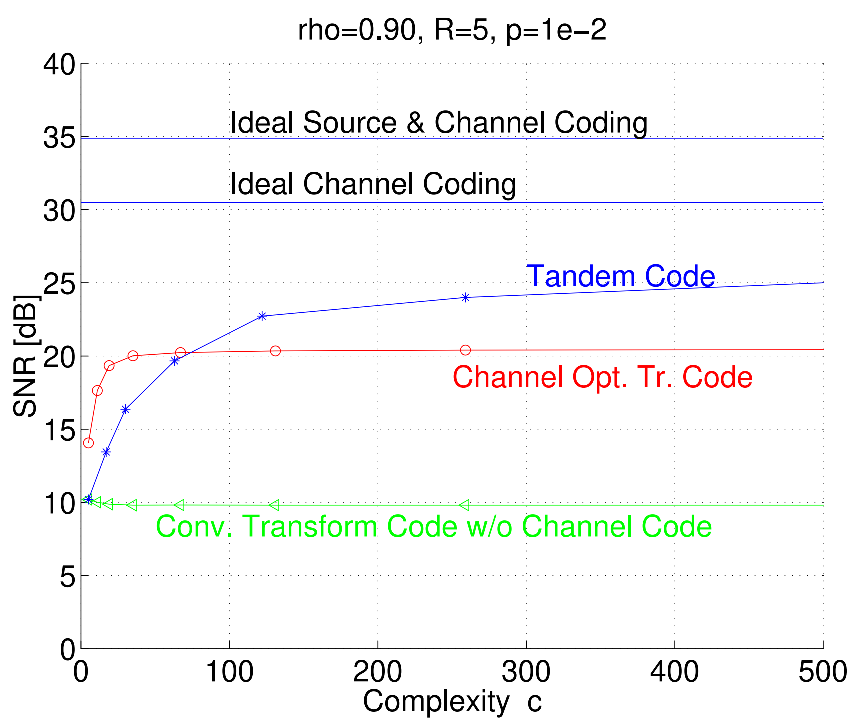


(c)

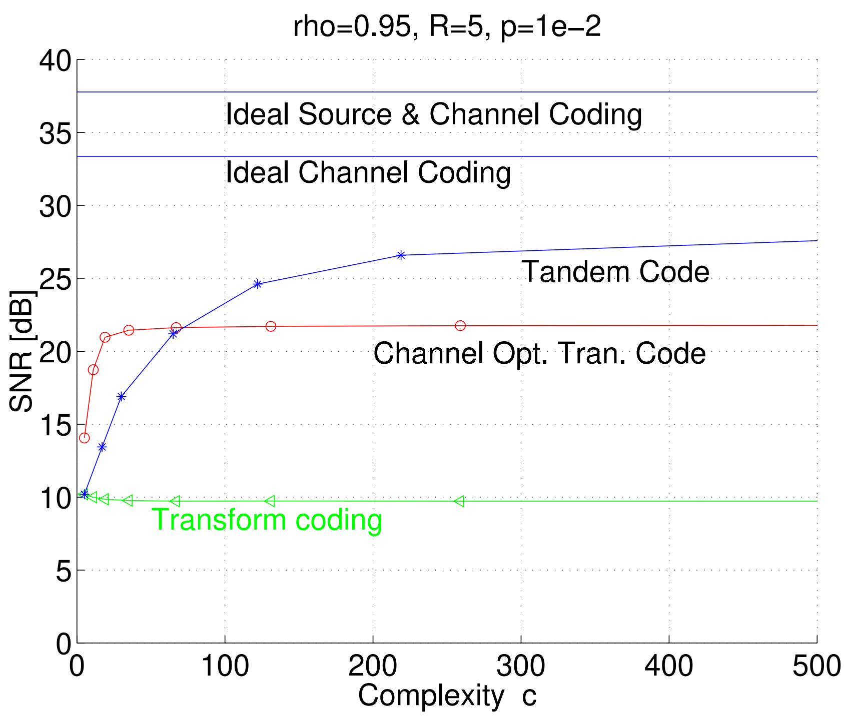
Figure 3.10: A comparison between channel-optimized transform coding and tandem source-channel coding for p=0.01 and $\rho = 0.9$. (a) R=3 (b) R=4 (c) R=5


rho=0.70, R=5, p=1e−2
40
35
30
25
20
15
10
5
0
0
100
200
300
400
500
SNR [dB]
Complexity c
Ideal Source & Channel Coding
Ideal Channel Coding
Tandem Code
Channel Opt. Tran. Code
Transform coding

(a)


rho=0.90, R=5, p=1e−2
40
35
30
25
20
15
10
5
0
0
100
200
300
400
500
SNR [dB]
Complexity c
Ideal Source & Channel Coding
Ideal Channel Coding
Tandem Code
Channel Opt. Tr. Code
Conv. Transform Code w/o Channel Code

(b)



(c)

Figure 3.11: A comparison between channel-optimized transform coding and tandem source-channel coding for p=0.01 and R=5. (a) $\rho = 0.70$ (b) $\rho = 0.90$ (c) $\rho = 0.95$

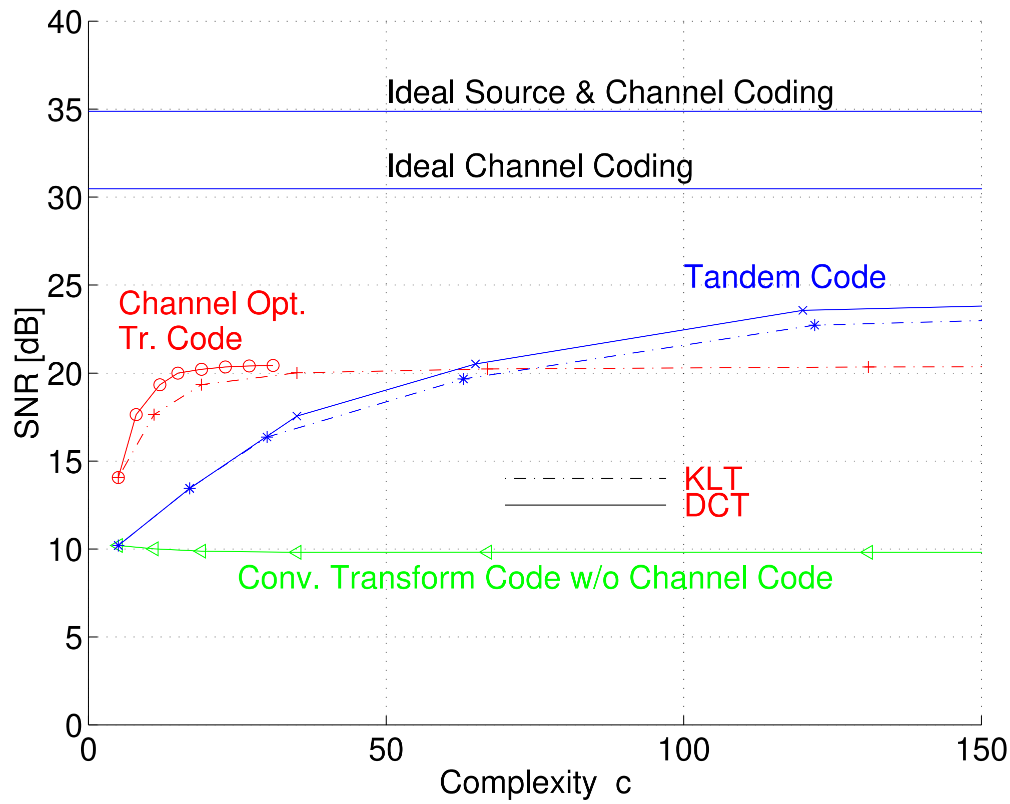


Figure 3.12: The performance of channel-optimized transform coding and tandem coding with DCT and KLT for R=5, $p=10^{-2}$ and $\rho=0.9$

TABLE 3.2
Complexity thresholds and gains of tandem over joint coding
at 2 times threshold for $\rho = 0.9$

		R 3	4	5
	0.02	200 / 0.2 dB	180 / 0.9 dB	115 / 1.5 dB
p	0.01	90 / 0.7 dB	90 / 1.5 dB	80 / 2.6 dB
	0.001	25 / 0.7 dB	35 / 1.5 dB	35 / 3.1 dB

TABLE 3.3
Complexity thresholds and gains of tandem over joint coding
at 2 times threshold for R=5

		ρ 0.70	0.90	0.95
	0.02	90 / 0.6 dB	115 / 1.5 dB	105 / 2.6 dB
p	0.01	50 / 0.9 dB	80 / 2.6 dB	75 / 3.2 dB
	0.001	25 / 2.9 dB	35 / 3.1 dB	35 / 3.3 dB

CHAPTER IV
UNEQUAL ERROR PROTECTION IN TANDEM SOURCE-CHANNEL CODING WITH COMPLEXITY CONSTRAINTS

4.1 Introduction

In the previous chapter, transform coefficients of the tandem source-channel coding are encoded into one Reed-Solomon channel code. Thus each coefficient is protected with the same level of protection. However, the transform coefficients have different levels of importance in the sense that the distortion caused by a bit error in one coefficient is somewhat different from that caused by a bit error in another coefficient. As previously discussed, unequal error protection is an approach that utilizes this fact. In transform coding systems, unequal error protection employs several channel codes with different protection levels for the coefficients with different levels of importance.

In this chapter, we apply the unequal error protection scheme to the previous tandem source-channel coding system with a complexity constraint. We consider an unequal error protection system consisting of an L-dimensional transform and L RS codes with different channel code rates. Given a complexity constraint, we try to optimize the performance of the tandem source-channel coding system by choosing the optimal channel code rates of the RS codes.

4.2 Unequal Error Protection System

Figure 4.1 shows an unequal error protection system consisting of an L-dimensional KL transform and L RS codes. The basic operation of the KLT and the scalar quantizers is the same as that of the previous tandem source-channel coding system. But each quantized coefficient is encoded with a different RS codes.

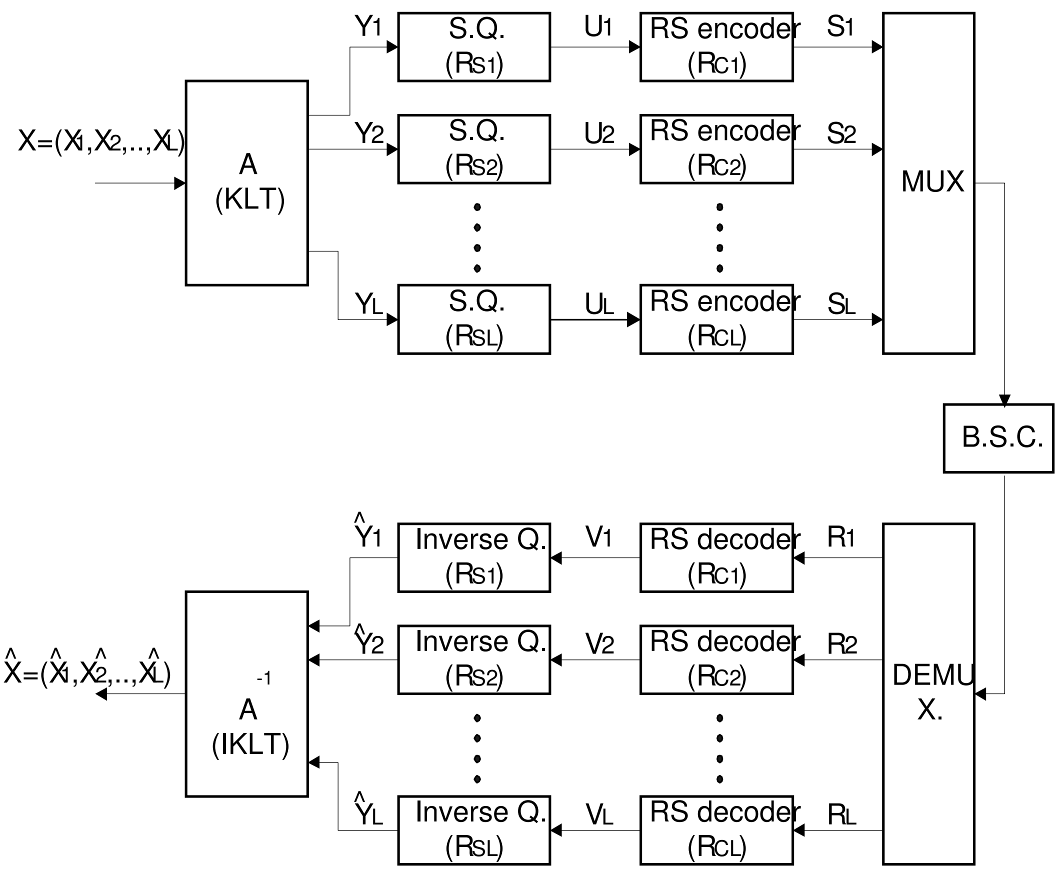


Figure 4.1: Unequal error protection system consisting of an L dim. KL transform and L RS codes

Let $R_{S,i}$ be the source code rate of ith transform coefficient, $R_{C,i}$ be the channel code rate of ith RS code, and R_i be the number of bits allocated for transmitting the ith coefficient. For a given bit allocation, R_1, R_2, ... , R_L, the average transmission rate R_{av} in number of bits per source sample is described by

$$R_{av} = \frac{1}{L}\sum_{i=1}^{L} R_i = \frac{1}{L}\sum_{i=1}^{L}\frac{R_{S,i}}{R_{C,i}}. \tag{4.1}$$

4.3 Optimization

For the system described in the previous section, we wish to minimize the mean-squared error

$$D = \frac{1}{L}\sum_{i=1}^{L} E\left(X_i - \hat{X}_i\right)^2 \tag{4.2}$$

subject to a constraint on the average transmission rate, the number of bits per source sample, i.e.,

$$R_{av} = \frac{1}{L}\sum_{i=1}^{L} R_i = \frac{1}{L}\sum_{i=1}^{L}\frac{R_{S,i}}{R_{C,i}} \leq R, \tag{4.3}$$

and subject to a complexity constraint, C, i.e.,

$$C_{tr} + \frac{1}{L}\sum_{i=1}^{L} C_i \leq C, \tag{4.4}$$

where C_{tr} is $4L - 2$, the complexity required by the transform, and C_i is the complexity given to ith quantizer and RS coder. Since the KLT is orthogonal, the distortion becomes

$$D = \frac{1}{L}\sum_{i=1}^{L} E\left(Y_i - \hat{Y}_i\right)^2. \tag{4.5}$$

The ith coefficient distortion, $E\left(Y_i - \hat{Y}_i\right)^2$, is a function of R_i, C_i and the error probability of the binary symmetric channel p. Let $D_i^*(R_i, C_i)$ denote the minimum value of the ith transform coefficient distortion for the given binary symmetric channel over all choices of scalar quantizer and RS code that satisfy the rate constraint R_i and the complexity constraint C_i. Then the minimum overall distortion given a bit allocation $R_1, R_2, \ldots, R_L$, and a complexity allocation $C_1, C_2, \ldots, C_L$ is

$$D^*(R_1, \cdots, R_L, C_1, \cdots, C_L) = \frac{1}{L}\sum_{i=1}^{L} D_i^*(R_i, C_i). \tag{4.6}$$

In order to minimize the overall distortion, we now need to find an optimal bit allocation and a complexity allocation satisfying the constraints in (4.3) and (4.4). The overall minimum distortion is

$$D^*(R, C) = \min_{\frac{1}{L}\sum_{i=1}^{L} R_i \le R,\ \frac{1}{L}\sum_{i=1}^{L} C_i \le (C - C_{tr})} D^*(R_1, \cdots, R_L, C_1, \cdots, C_L). \tag{4.7}$$

Thus, our optimization procedure should be performed in three steps: (1) given L, R_1, R_2, ... , R_L, and C_1, C_2, ... , C_L satisfying (4.3) and (4.4), for each i, find the scalar quantizers Q_i and the RS coders RS_i, to be applied to the ith coefficient, (2) determine the optimal bit allocation and the complexity allocation that satisfy the constraints in (4.3) and (4.4) and that minimize the distortion for a given L, and (3) find the transform size L to minimize the overall distortion, where each system for a given L is optimized through the Steps (1) and (2). This last step is performed with the values of L coming from the set {1, 2, 4, 8, ... , 128}.

In the Step (1), we find $D_i^*(R_i, C_i)$ for $i = 1, 2, \ldots, L$; $R_i = 0, 1, \ldots, R_{max}$; and $C_i = C_0, 2C_0, 3C_0, \ldots, C_{max}$, where C_0 is a positive integer. This optimization can be done with the previous optimization method of Chapter 3 for the tandem source-channel coding system with $L = 1$.

Step (2) requires the allocation of two resources: rate and complexity. For a given L, R and C, we need to allocate LR bits and LC operations among L coefficients. Because there are so many parameters to allocate, this is a very complicated task. Because of this, we take a supoptimal approach and consider an allocation algorithm that has a relationship

$$C_i = \frac{C - C_{tr}}{R} R_i , \tag{4.8}$$

i.e., allocating the complexity proportional to the rate. It satisfies the complexity constraint in (4.4):

$$\begin{aligned} C_{tr} + \frac{1}{L}\sum_{i=1}^{L} C_i &= C_{tr} + \frac{1}{L}\sum_{i=1}^{L} \frac{C - C_{tr}}{R} R_i \\ &= C_{tr} + \left(C - C_{tr}\right)\frac{1}{R}\frac{1}{L}\sum_{i=1}^{L} R_i \le C_{tr} + \left(C - C_{tr}\right)\frac{1}{R} R = C, \end{aligned} \tag{4.9}$$

where the inequality comes from the rate constraint (4.3). Now, the optimization is reduced to finding the distortion:

$$D(R, C) = \min_{\frac{1}{L}\sum_{i=1}^{L} R_i \le R} D^*(R_1, \cdots, R_L, C_1 = \frac{C - C_{tr}}{R} R_1, \cdots, C_L = \frac{C - C_{tr}}{R} R_L) . \tag{4.10}$$

Thus, the problem of allocating two resources is reduced to finding just the bit allocation R_1, R_2, ... , R_L satisfying the rate constraint (4.3). Although this algorithm may not provide the minimum distortion, we use it since it is simple and since no algorithm guaranteeing the optimal performance is known. The problem of finding the best allocation of rate and complexity will be a good topic for future research. To find the bit allocation described by (4.10), we use the conventional bit allocation method [32, pp. 234 - 235].

4.4 Results and Conclusion

A representative result for the optimized unequal error protection system is shown in Figure 4.2, which plots the performance of the previous tandem source-channel coding system, the channel optimized transform coding system and the unequal error protection system, all for $R = 5$ and $p = 10^{-2}$. The unequal error protection system gets better performance than the previous tandem source-channel coding system. Specifically, when the complexity constraint is between 50 and 100, the gain of the unequal error protection system over the equal error protection system is about 2 dB. The gain decreases as the complexity constraint gets larger. Because of the gain for low value of complexity, the complexity threshold between tandem source-channel coding and channel-optimized transform coding, which is discussed in the previous chapter, becomes smaller.

In summary, the unequal error protection scheme improves the performance of tandem source-channel coding system, especially for small complexity constraints. Thus it makes the complexity threshold smaller than that of the equal error protection system. If a more sophisticated algorithm for allocating rate and complexity is designed, the improvement of unequal error protection systems will increase, and the threshold will be even more smaller.

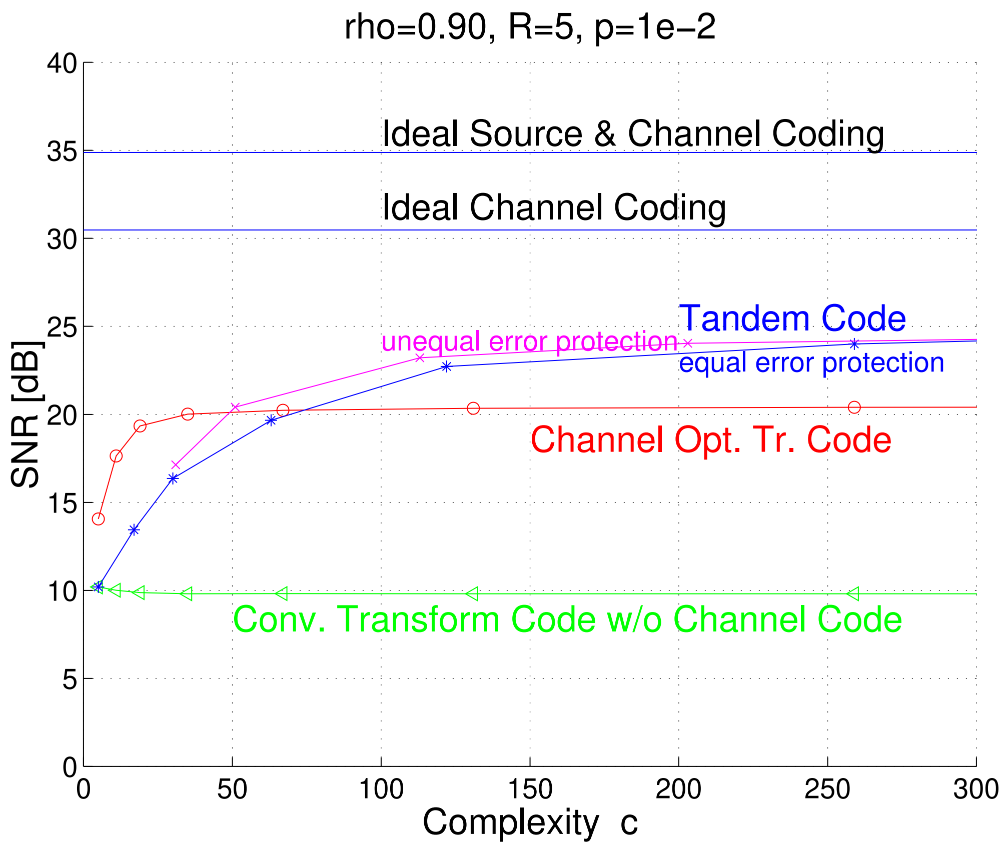


Figure 4.2: The performance comparison between the previous tandem source-channel coding system, channel optimized transform coding system and unequal error protection system for $R = 5$ and $p = 10^{-2}$

CHAPTER V
JOINT AND TANDEM SOURCE-CHANNEL CODING WITH DELAY CONSTRAINTS

In this chapter, we compare two common source-channel coding strategies, joint and tandem source-channel coding, on the basis of distortion vs. delay by analyzing specific representatives of each when transmitting analog data samples across a binary symmetric channel. For the representative systems of joint and tandem source-channel coding which were considered in the previous chapter, delay formulas are derived and then used to optimize the performance of both systems with a given delay constraint. The results follow to show the comparison of both systems on the basis of distortion vs. delay.

5.1 Introduction

As discussed earlier, Shannon's separation theorem cannot be applied to applications where the system has complexity constraints or/and delay constraints. This limitation has motivated the use of joint source-channel. In Chapter 3, we have looked for evidence for the claim that joint source-channel coding can achieve better performance with less complexity than tandem source-channel coding, by quantitatively comparing representative systems of each type on the basis of distortion vs. complexity. In this chapter, we expand our investigation to find whether joint source-channel coding achieves better performance with less delay than tandem source-channel coding. For this purpose, we compare joint and tandem source-channel coding on the basis of distortion vs. delay by analyzing specific representative systems.

We consider the representative systems that were considered in Chapter 3. The delay of each system is derived and expressed in units of data samples by considering buffering delay caused by block processing. Using the distortion analyzed in Chapter 3, both systems are optimized under a delay constraint.

5.2 The Systems

As representatives of joint and tandem source-channel coding system, we use the systems of Chapter 3. These are summarized as follows.

The source is a stationary Gauss-Markov random sequence and the channel is a binary symmetric channel. There are R channel uses per source sample. As the joint source-channel code, we consider the channel-optimized transform code studied by Vaishampayan and Farvardin [5]. As the tandem source-channel code, we consider a conventional transform code (optimized for a noiseless channel) followed by a Reed-Solomon (RS) channel code. Section 3.2 has the detailed system description. The distortion of each type of system is analyzed in Section 3.3.

5.3 The Analysis of Delay

In this section, we analyze the delay of each system. The delay introduced by a system when transmitting source symbol U_i is basically the number of data samples U_{i+1}, U_{i+2},.... that subsequently arrive at the encoder until the time that the reproduction $\hat{U}_i$ is produced by the decoder. For blockwise processing, which is common in most source-channel coding systems, the delay may vary with each source sample. For example, the first source sample in a block cannot be transmitted until the last source sample is available to process, whereas the last sample is immediately ready to process and transmit. In this study, we consider the delay of a source-channel code as the maximum delay for any source symbol.

Many factors, such as buffering, processing and transmission, cause delay in various ways in the communication systems. Though delay can be caused by encoding/decoding hardware and transmission, we consider only buffering delay because it is always present and it is the only type of delay that cannot be reduced by adding resources.

5.3.1 The Delay of Channel-Optimized Transform Codes

We consider only buffering delay by assuming that the transform encoding/decoding and transmission are instantaneously performed and thus the reconstructed source samples are available when L source samples are ready to be encoded. Thus, we take the delay for channel-optimized transform codes to be simply the dimension L of its transform, which is the time required to buffer the L samples of one block. The delay of a channel-optimized transform code is

$$\tau_J = L. \tag{5.1}$$

5.3.2 The Delay of Tandem Codes

Like the channel-optimized transform code, for the delay of tandem source-channel codes we consider only buffering delay by assuming that the transform encoding/decoding, the RS encoding/decoding and transmission are instantaneously performed. For the analysis of delay in a tandem system, we consider the encoding of M_sL data samples as described in Chapter 3. That is, we have

$$M_sLR_S = M_Ckm, \tag{5.2}$$

where M_s and M_c do not have any common divisor except 1. By dividing both sides in (5.2) with M_sM_c, we have

$$\frac{LR_S}{M_C} = \frac{km}{M_S}. \tag{5.3}$$

Let the quantity of (5.3) be g. That is,

$$g = \frac{LR_S}{M_C} = \frac{km}{M_S}. \tag{5.4}$$

From (5.4), we have $LR_s = gM_c$ and $km = gM_s$, which imply that g is the greatest common divisor of LR_s and km since M_s and M_c do not have any common divisor except 1 as said earlier.

We now take g bits as the "packing unit" for our packing problem, where M_sLR_s bits from the M_s transform codes are packed into M_c channel codewords. Then, LR_s transform code output bits consist of M_c units since $LR_s = gM_c$, and one channel codeword accepts M_s units since $km = gM_s$. Since $M_sLR_s = M_sgM_c = gM_sM_c$, our packing problem becomes the problem of packing M_sM_c units coming from M_s transform blocks into M_c codewords. An example for $M_s = 5$ and $M_c = 3$ is described in Figure 5.1.

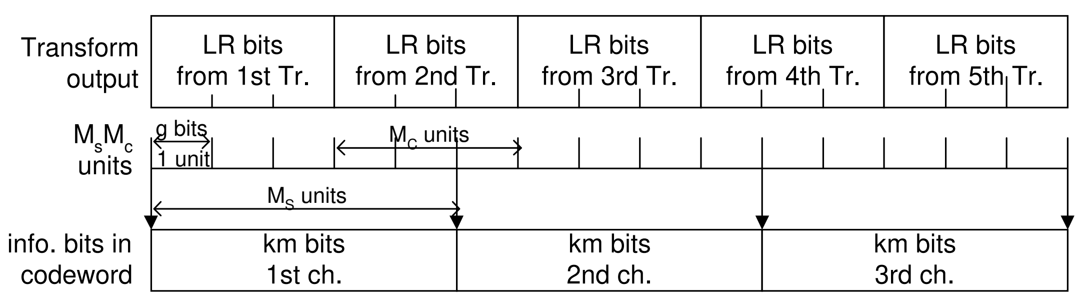


Figure 5.1: Packing M_S transform outputs into M_C channel codewords ($M_S = 5$, $M_C = 3$)

The appropriate expression for delay depends on the values of LR_s and km. Thus we need to consider four cases: (1) km is a multiple of LR_S, (2) km is not a multiple of LR_S, but is greater than LR_S, (3) LR_S is a multiple of km, and (4) LR_S is not a multiple of km and LR_S is greater than km.

(1) The case that km is a multiple of LR_S

If km is a multiple of LR_S, i.e., $M_S LR_S = km$, then M_S transform outputs are nicely packed into one codeword. An example for $M_S = 2$ is shown in Figure 5.2. Thus, the delay is the time required for buffering M_S transform outputs for channel encoding, i.e.

$$\tau_T = M_S L = (km/LR_S)L. \tag{5.5}$$

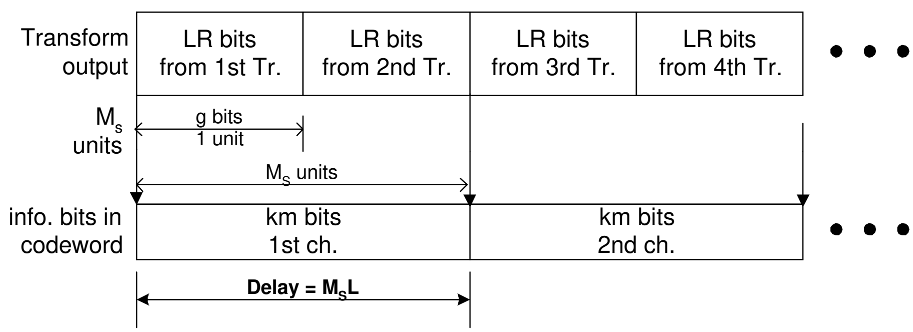


Figure 5.2: Packing transform outputs into channel codewords ($M_S = 2$, $M_C = 1$)

(2) The case that km is not a multiple of LR_S, but is greater than LR_S

If km is not a multiple of LR_S, but is greater than LR_S, then one channel codeword, accepting M_S units, takes M_S units from more than one transform block since $km > LR_S$, i.e. $M_S > M_C$. An example for $M_s = 5$ and $M_c = 3$ is shown in Figure 5.3 (a). For a given R-S codeword, let M_T denote the number of transform blocks from which it receives

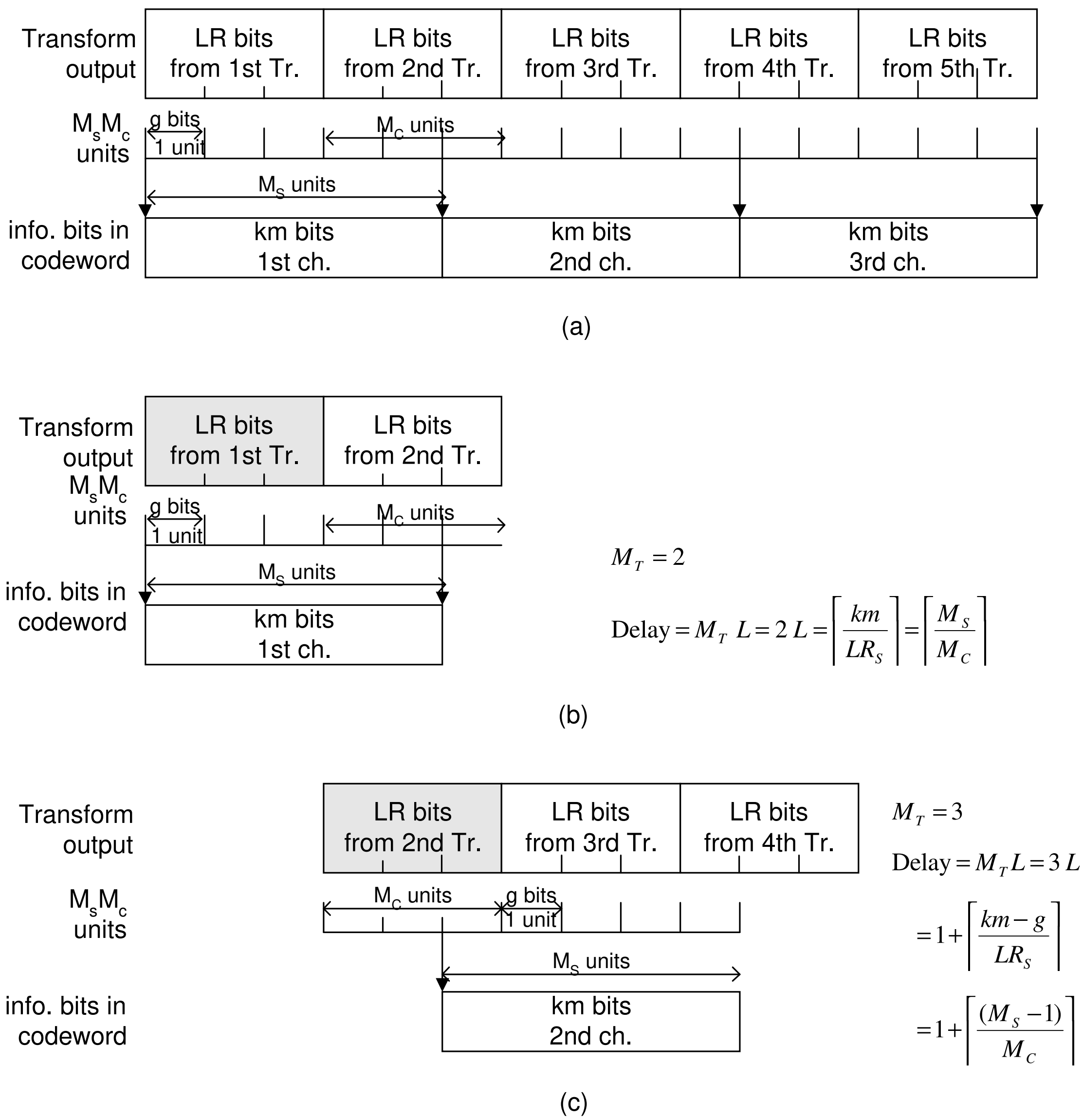


Figure 5.3: The delay for $M_s > M_c$ in tandem coding system ($M_s = 5$, $M_c = 3$) (a) packing transform outputs into channel codewords, (b) M_T for the first channel codeword and delay for the first transform block, (c) M_T for the second channel codeword and delay for the second transform block

units. M_T is the number of transform block outputs that need to be buffered, for packing M_S units into the ith codeword. Since $km > LR_S$, the minimum number of transform blocks that one channel codeword takes is $\lceil km/LR_S \rceil = \lceil M_S/M_C \rceil$ as shown in Figure 5.3.(b). We have the maximum value of M_T when the codeword takes its first unit as the last unit of one transform code outputs and takes the other (M_S – 1) units from the subsequent $\lceil (M_S - 1)/M_C \rceil$ transform code outputs. Thus, M_T for the codeword is 1 + $\lceil (M_S - 1)/M_C \rceil$ as shown in Figure 5.3.(c), where the second codeword takes 1 unit from the second transform code outputs and the other (M_S – 1) units (4 units) from the third and fourth transform code outputs. Since the maximum value of M_T is 1 + $\lceil (M_S - 1)/M_C \rceil = 1 + \lceil (km - g)/LR_S \rceil$, the delay for the case (2) is

$$\tau_T = \left(1 + \left\lceil \frac{km - g}{LR_S} \right\rceil\right) L. \tag{5.6}$$

(3) The case that LR_S is a multiple of km

For the case that LR_S is a multiple of km, the transform outputs are nicely packed into M_c codewords as shown in Figure 5.4. Thus the delay is the time required for buffering source samples for transform encoding and is given by

$$\tau_T = L. \tag{5.7}$$

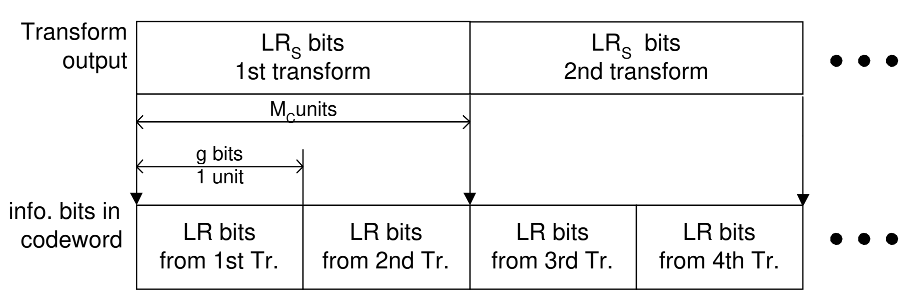


Figure 5.4: Packing transform outputs into channel codewords ($M_S = 1$, $M_C = 2$)

(4) The case that LR_S is not a multiple of km and LR_S is greater than *km*

If LR_S is not a multiple of *km* and LR_S is greater than *km*, then the M_c units coming from one transformation are divided and packed into more than one channel codeword since $M_S < M_c$. Figure 5.5 give an example for $M_s = 3$ and $M_c = 5$. Thus, the first unit from a given transform output and the last are to be packed into separate codewords. The codeword receiving the last unit of the given transformation cannot be sent right after the transformation since it is not full and thus it needs to wait for some units from the next transform to fill the rest part of the codeword. When the codeword containing the last unit of the transformation is decoded, all bits resulting from the prior transformation are available and the source decoding for the prior transformation can be performed. Thus, there is need to buffer two transform blocks and thus the delay is

$$\tau_T = 2\,L. \tag{5.8}$$

Considering the above four cases, the delay of tandem coding system is summarized as follow.

$$\tau_T = \begin{cases} (km/LR_S)L, & \text{if } km \text{ is a multiple of } LR_S \\ \left(1+\left\lceil \dfrac{km-g}{LR_S} \right\rceil\right)L, & \text{if } km \text{ is not a multiple of } LR_s\text{, but } km > LR_S \\ L, & \text{if } LR_S \text{ is a multiple of } km \\ 2L, & \text{if } LR_S \text{ is not a multiple of } km\text{, but } LR_S > km. \end{cases} \tag{5.9}$$

However, a little checking will show that the delay expressions for the four cases can be combined into:

$$\tau_T = \left(1+\left\lceil \frac{km-g}{LR_S} \right\rceil\right)L, \tag{5.10}$$

where g is the greatest common divisor of *km* and LR_S.

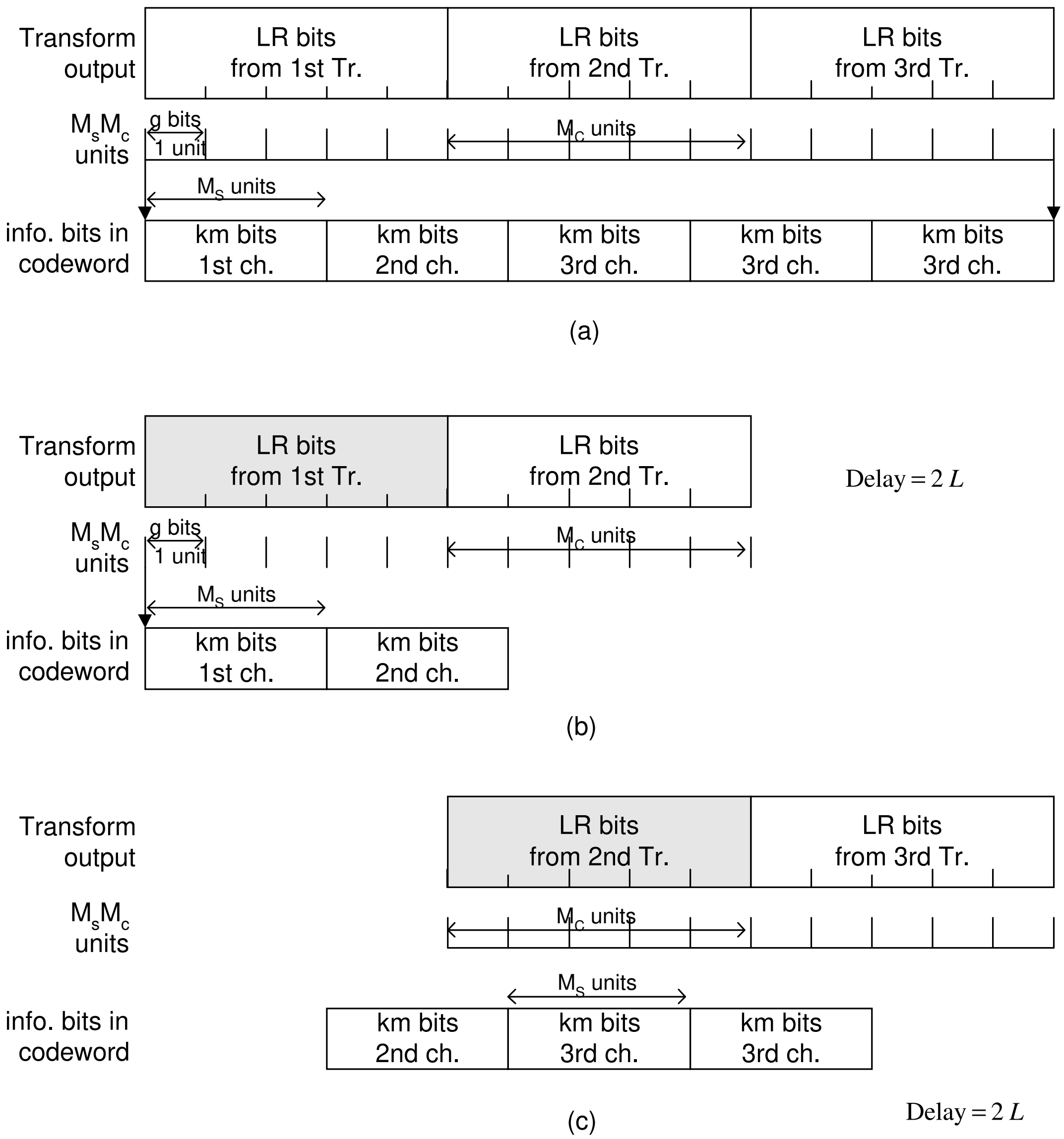


Figure 5.5: The delay for $M_s < M_c$ in tandem coding system (M_s =3, M_c=5) (a) packing transform outputs into channel codewords, (b) M_T for the first channel codeword and delay for the first transform block, (c) M_T for the second channel codeword and delay for the second transform block

5.4 Optimizing Distortion With A Delay Constraint

5.4.1 Channel-Optimized Transform Coding

The purpose of the optimization for the channel-optimized transform code is to find the transform parameter L that gives the minimum distortion under a delay constraint. Since the delay of the channel-optimized transform code is directly proportional to the transform dimension as shown in (5.1) and since the performance of the system improves with the transform dimension, the optimization procedure need only find the largest transform dimension satisfying the delay constraint and compute the distortion of the transform code with that transform dimension. Thus the optimization procedure is summarized as follows. For specific choices of the source parameter ρ, channel parameters R and p, and a delay constraint τ, a computer program evaluates the distortion of the transform code for the largest transform dimension L in the set $\{1,2,4,8,\ldots,128\}$ such that the delay given by (5.1) is no larger than τ.

5.4.2 Tandem Source-Channel Coding

For specific values of the source parameter ρ, channel parameters R and p, and a delay constraint τ, a computer program searches for the best choices of source code parameters L, R_S and channel code parameters n, k and m subject to the constraints $R_S \le R$, $nR_S / R \le k$, LR_S is an integer, $n = 2^m - 1$, $1 \le k < n$ and $\tau_T \le \tau$. To do so, the program uses the expressions of Section 3.3 and 5.3 to compute the MSE and delay with parameter values varying in three nested loops. The values of L come from the set $\{1,2,4,8,16,32,64,128\}$; the values of R_S are such that LR_S equals $1,2,\cdots,\lfloor RL \rfloor$; the values of m come from the set $\{1,\cdots,12\}$; and finally, $k = \lceil nR_S / R \rceil$.

5.5 Results and Conclusions

Figure 5.6 provides representative results of the optimizations under delay constraints. It plots the SNR in dB of channel-optimized and tandem source-channel coding for the case that the source has correlation coefficient $\rho = 0.9$, the BSC has error probability $p = 2\text{x}10^{-2}$, and the channel is used $R = 5$ times per data sample. The SNR of a tandem system consisting of conventional transform code with L=128 and an ideal channel code, and the best possible performance of any tandem code are shown in the figure. Also the performance of the conventional transform code without channel codes is shown. It is easy to see that there is no benefit to increasing the delay in the conventional transform code, i.e. the transform dimension. However, as delay increases, the SNR of channel-optimized transform coding improves. Specifically, there is about a 5 dB gain over conventional transform coding for $\tau = 4$, and the gain increases until it saturates at about 11 dB for $\tau \cong 30$. At this point, there is no benefit to further increases in dimension.

The performance of optimized tandem source-channel codes also increases with delay. For small values of delay, its SNR is not as large as that of channel-optimized transform coding, while for larger values of delay, it is larger. In particular, tandem coding becomes better at around 50 samples of delay. At this point, although the SNR of channel-optimized transform coding has saturated, allowing additional delay permits the tandem code to continue to improve. Thus, there appears to be a threshold such that channel-optimized transform coding gives better performance than tandem coding when and only when the delay is constrained to be less than this threshold.

We repeated the optimization for a variety of other choices for R, p and ρ. Figure 5.7 (a), (b) and (c) show the performance comparison between channel-optimized transform coding and tandem source-channel coding for $R = 5$, $\rho = 0.9$, and $p = 0.02$, 0.01 and 0.001, respectively. As shown in these figures, delay thresholds become smaller as p decreases. This trend is observed for other values of R. Tables 5.1 summarizes this trend by showing the thresholds and the gain of the tandem system over the joint system at 5 times the threshold. The latter is intended as an indicator of how much can be gained with the additionally permissible delay of tandem coding. We can see that the delay thresholds decrease and the gains increase as the channel becomes more reliable. Figure 5.8 shows the change of delay thresholds as R varies. It plots the performance of the joint and tandem systems for $\rho = 0.9$, $p = 0.01$, and $R = 3$, 4 and 5. The thresholds decreases as more channel uses are available. Specifically, the threshold is 125 for $R = 3$, 55 for $R = 4$, and 25 for $R = 5$. Also, the gains increases as R increases. The detailed gains are shown in Tables 5.1. This trend on the thresholds and gains means that contrary to what one might initially think, channel coding is more immediately useful, i.e. useful with moderate delay, when the channel is more reliable.

Figure 5.9 shows the variation of delay thresholds as ρ varies by plotting the performance of joint and tandem systems for $p = 0.01$, $R = 5$, and $\rho = 0.70$, 0.90 and 0.95. Table 5.2 shows the delay thresholds and the gain of the tandem system over the joint system at 5 times the threshold for $R = 5$, various p's and various ρ 's. The thresholds tend to decrease and the gains increase as the source becomes more correlated.

In summary, the distortion vs. delay performance of two common source-channel coding strategies, joint and tandem source-channel coding, has been computed and compared by analyzing specific representative systems. The results indicate that there exists a threshold in delay such that when the permissible delay is above the threshold, tandem coding is better, and when below the threshold, channel-optimized transform coding is better. The results suggest using joint source-channel coding when low delay is required, and using tandem coding when better performance is needed and substantially more delay is allowed.

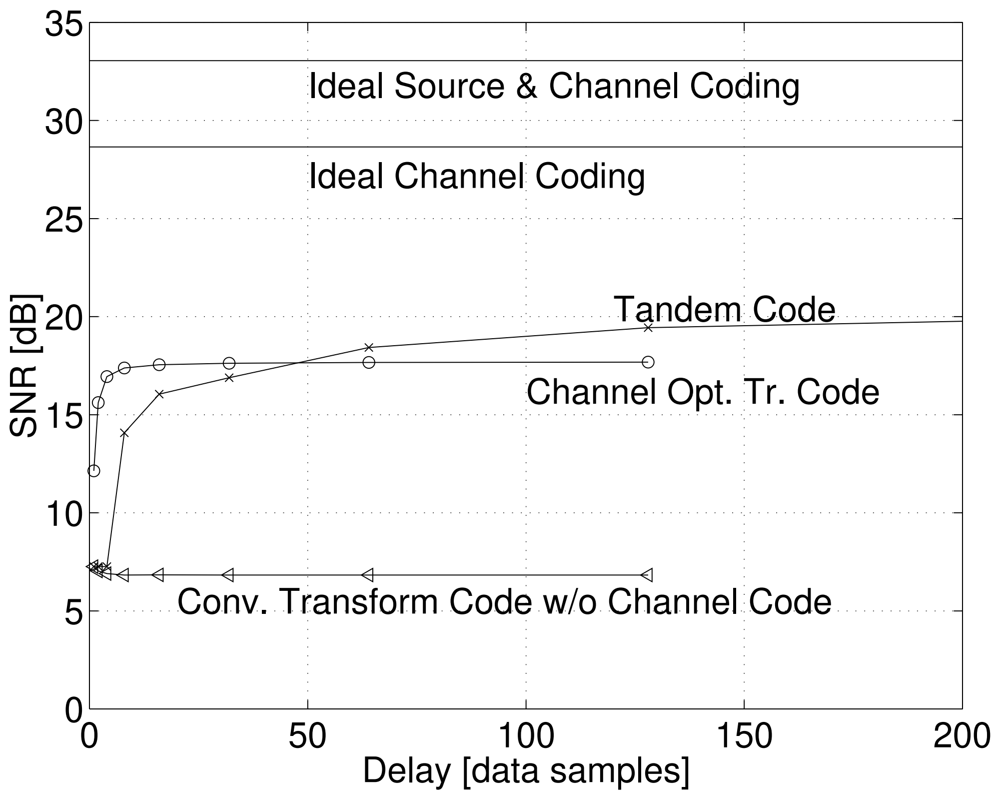
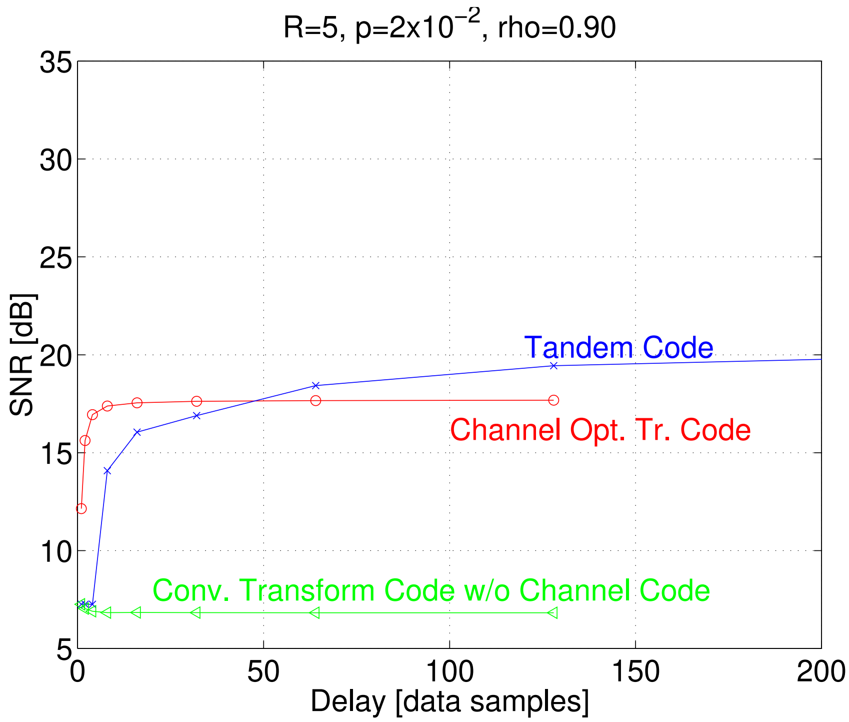
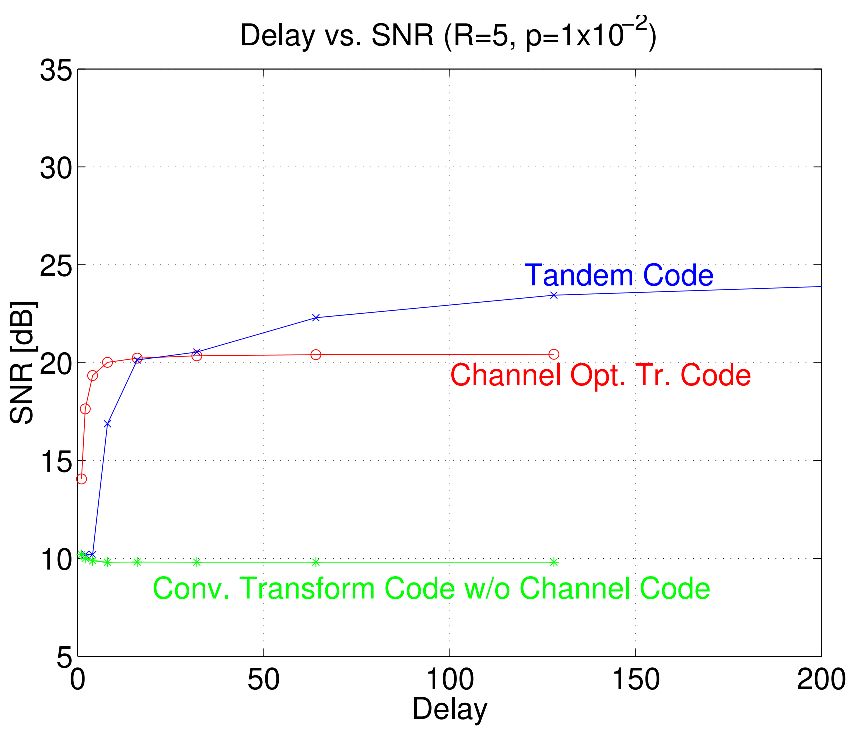


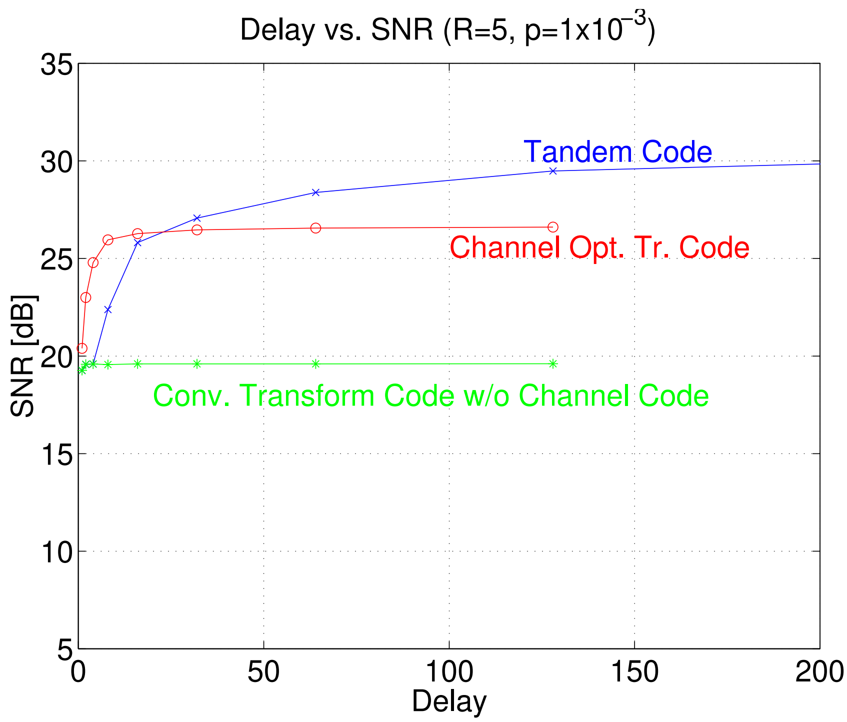
Figure 5.6: Performance of channel-optimized transform coding and tandem coding for R=5, $p=2x10^{-2}$ and $\rho = 0.9$


R=5, p=2x10^−2, rho=0.90
SNR [dB]
Delay [data samples]
Tandem Code
Channel Opt. Tr. Code
Conv. Transform Code w/o Channel Code
35
30
25
20
15
10
5
0
50
100
150
200

(a)

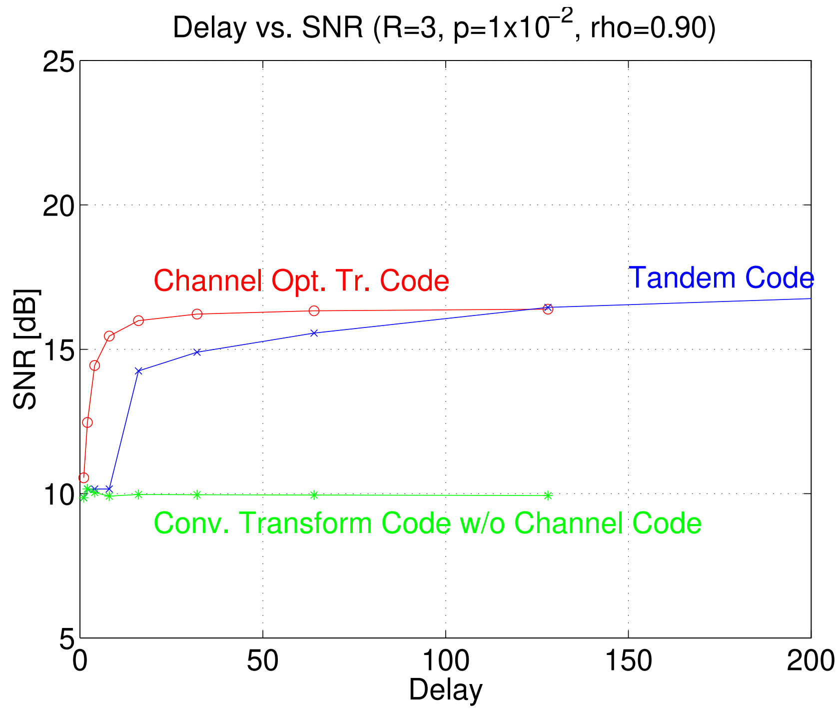

Delay vs. SNR (R=5, p=1x10^−2)
SNR [dB]
Delay
Tandem Code
Channel Opt. Tr. Code
Conv. Transform Code w/o Channel Code
35
30
25
20
15
10
5
0
50
100
150
200

(b)

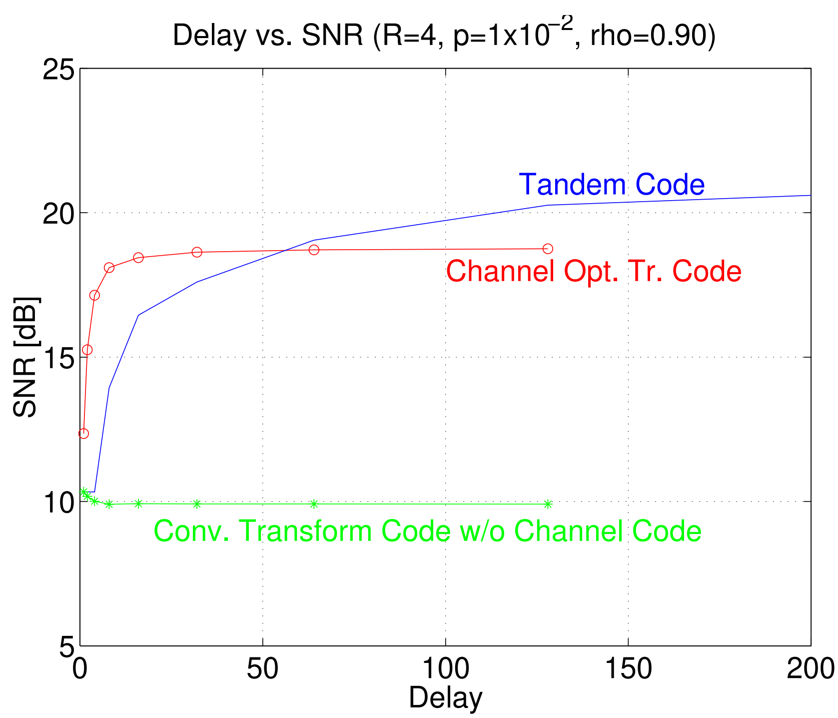


(c)

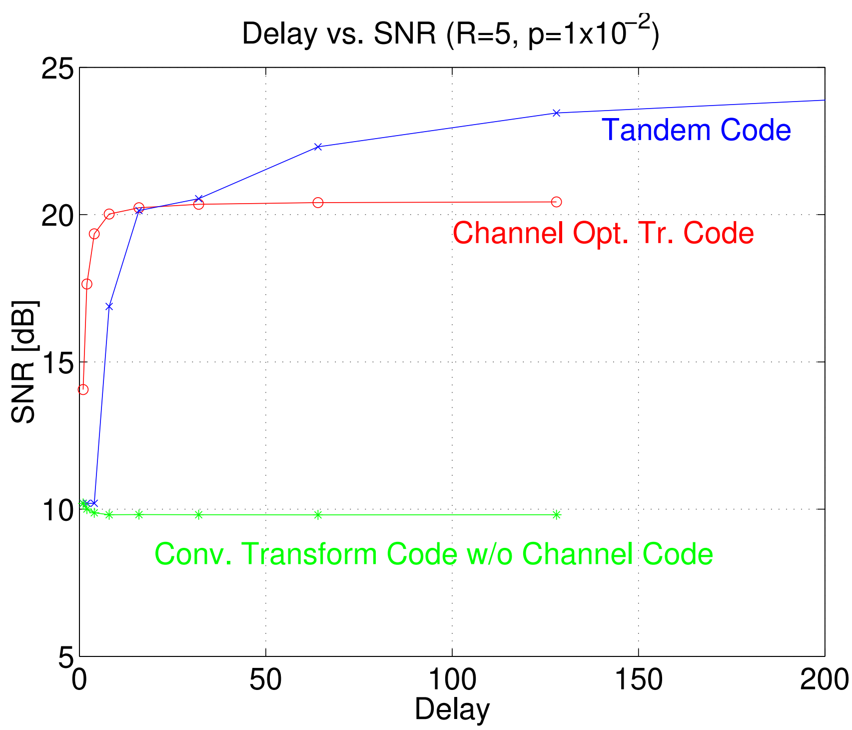
Figure 5.7: A comparison between channel-optimized transform coding and tandem source-channel coding for R=5 and $\rho = 0.9$. (a) p=0.02 (b) p=0.01 (c) p=0.001


Delay vs. SNR (R=3, p=1x10^{-2}, rho=0.90)
SNR [dB]
Delay
Channel Opt. Tr. Code
Tandem Code
Conv. Transform Code w/o Channel Code
25
20
15
10
5
0
50
100
150
200

(a)

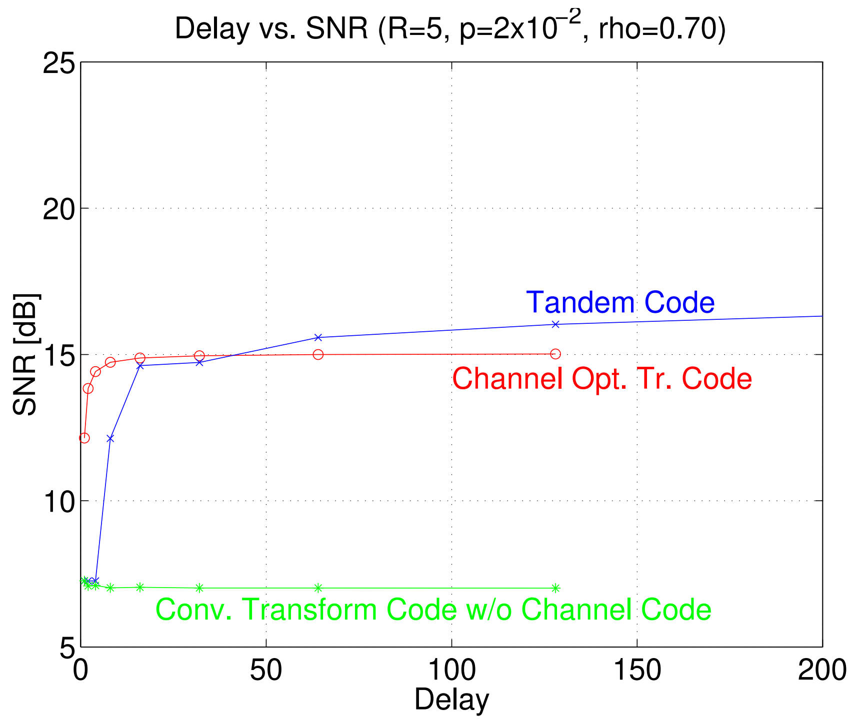

Delay vs. SNR (R=4, p=1x10^{-2}, rho=0.90)
SNR [dB]
Delay
Tandem Code
Channel Opt. Tr. Code
Conv. Transform Code w/o Channel Code
25
20
15
10
5
0
50
100
150
200

(b)



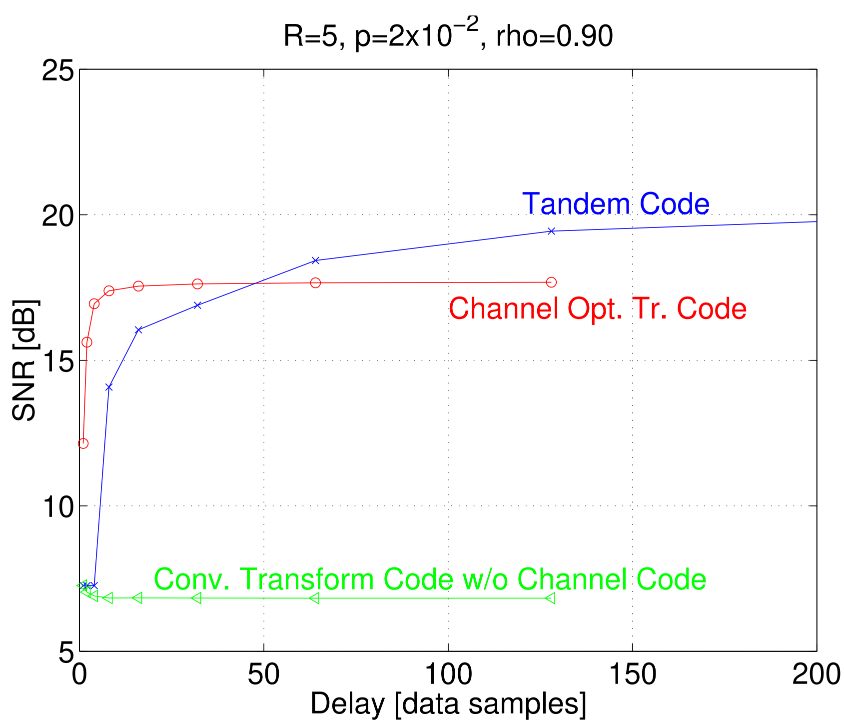
(c)

Figure 5.8: A comparison between channel-optimized transform coding and tandem source-channel coding for p=0.01 and $\rho = 0.9$. (a) R=3 (b) R=4 (c) R=5



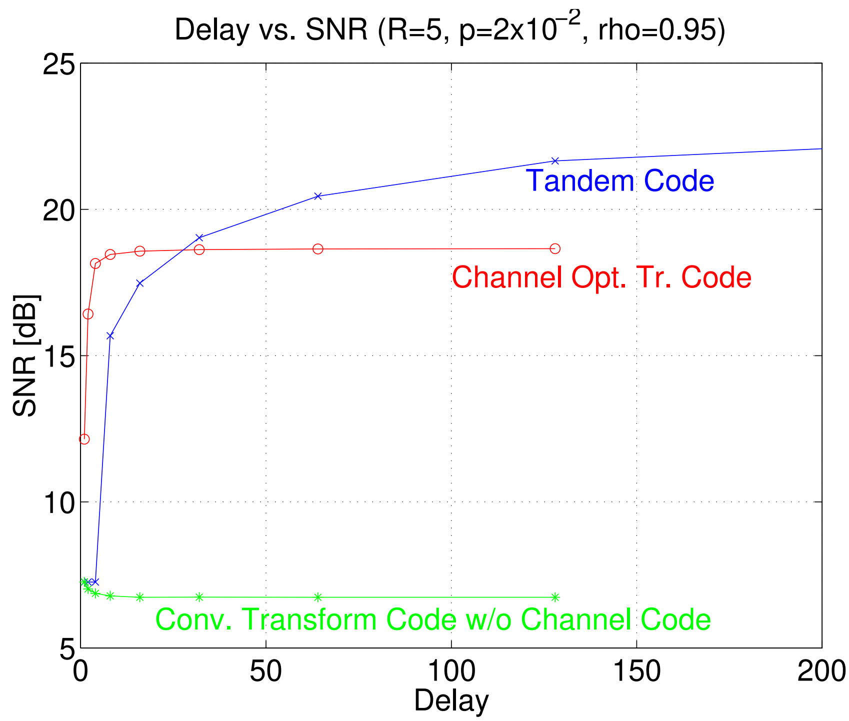
Delay vs. SNR (R=5, p=2x10^-2, rho=0.70)
SNR [dB]
Delay
Tandem Code
Channel Opt. Tr. Code
Conv. Transform Code w/o Channel Code
25
20
15
10
5
0
50
100
150
200

(a)



R=5, p=2x10^-2, rho=0.90
SNR [dB]
Delay [data samples]
Tandem Code
Channel Opt. Tr. Code
Conv. Transform Code w/o Channel Code
25
20
15
10
5
0
50
100
150
200

(b)



(c)

Figure 5.9: A comparison between channel-optimized transform coding and tandem source-channel coding for p=0.01 and R=5. (a) $\rho = 0.70$ (b) $\rho = 0.90$ (c) $\rho = 0.95$

TABLE 5.1
Delay thresholds and gains of tandem over joint coding
at 5 times threshold for $\rho = 0.9$.

		R		
		3	4	5
	0.02	400 / 0.6dB	80 / 0.8dB	47 / 1.9dB
p	0.01	125 / 0.7dB	55 / 2.1dB	25 / 2.5dB
	0.001	115 / 0.7dB	60 / 1.8dB	25 / 2.4dB

TABLE 5.2
Delay thresholds and gains of tandem over joint coding
at 5 times threshold for R=5

		ρ		
		0.70	0.90	0.95
	0.02	40 / 1.2dB	47 / 1.9dB	30 / 2.9dB
p	0.01	20 / 2.3dB	25 / 2.5dB	15 / 2.4dB
	0.001	15 / 1.8dB	25 / 2.4dB	30 / 3.2dB

CHAPTER VI
RATE ALLOCATION FOR NOISY CHANNELS

In this chapter, we consider the problem of rate allocation in transform codes for noisy channels. The rate allocation for transform codes for noisy channels is analyzed using high-resolution quantization theory. Then theoretically and numerically determined bit allocations are presented and compared.

6.1 Introduction

As discussed in Section 2.3, in designing optimal transform codes we encounter the problem of choosing transform dimension, the transform matrix and the scalar quantizers. For stationary Gaussian sources, and more generally for most sources, the performance improves with increasing transform dimension. However, in the real world, the transform size is often set by system constraints such as complexity and delay. The transform size, 8, is widely used in present image and video coding standards such as JPEG, H.261 and MPEG. Even though the Karhunen-Loeve Transform (KLT) is known as the optimal transform for the Gaussian source, the discrete-cosine transform (DCT) is widely used in many applications, including image and video coding, because it has faster computational algorithms, good performance and source independent design. Thus, in many cases the problem of optimal transform design condenses to the problem of finding the scalar quantizers with which to encode the transform coefficients. Since design algorithms for optimizing a scalar quantizer for a given rate are well known and understood, the remaining design step is choosing the rate of each coefficient's quantizer under an overall average rate constraint. This problem is called a bit or rate allocation, and several methods of optimal bit allocation are discussed in Section 2.3.1.

For noiseless channels, bit allocation has been well understood for a long time. There are numerical algorithms for optimizing bit allocation and there are high-resolution formulas that predict the optimal bit allocation. For noisy channels, Vaishampayan and Farvardin [5] have developed a numerical algorithm for optimizing transform codes, which in effect optimizes the bit allocation. However, there has been no high-resolution theory that predicts the optimal bit allocation. In this chapter, we use Hochwald and Zeger's recently developed high-resolution theory for noisy channel quantizers [26], to analyze the optimal bit allocation for transform codes for noisy channels.

The resulting asymptotic expression for the optimal bit allocation for noisy channels suggests that even more bits need to be allocated to the coefficients whose variances are greater than the geometric mean of the coefficient variances, compared to the bit allocation for noiseless channels. On the other hand, fewer bits are to be allocated to the coefficients whose variances are smaller than the geometric mean. However, as will be explained later, the Hochwald-Zeger theory applies to a kind of joint source-channel coding that does not exactly match the type of joint source-channel codes that we have so far studied and that is presumed by Vaishampayan and Farvardin's approach to noisy channel transform code design. Nevertheless, it is similar enough that we believe that the bit allocation based on the Hochwald-Zeger high-resolution theory makes a reasonably good prediction of the bit allocation found by the Vaishampayan-Farvardin method. Comparisons will be made later that confirm this.

6.2 High-Resolution Analysis

High-resolution quantization theory can provide asymptotic expressions for large rates for the distortions of optimal scalar quantizers. These approximate distortion expressions are used to express the overall distortion of the transform coding system. The resulting distortion can then be minimized under the average rate constraint using Lagrangian techniques or common inequalities.

6.2.1 Rate Allocation for noiseless channels

The optimal rate allocation for noiseless channels is discussed in Chapter 2. Here we review and summarize the well-known optimal rate allocation theory based on high-resolution quantization theory.

For noiseless channels, the least possible distortion of a scalar quantizer with rate R is

$$\delta_{sq}(R) \cong \sigma^2 \frac{1}{12} \beta 2^{-2R}, \text{ for large } R, \tag{6.1}$$

where σ^2 is the variance of the source and $\beta = 2\pi 3^{\frac{3}{2}}$ for a Gaussian source [34]. For a transform code with an L-dimensional transform T applied to a Gaussian source, the transform coefficients are Gaussian, and so with quantizers optimized for the coefficients, with rates $R_1,..., R_L$, the distortion is

$$\frac{1}{L}\sum_{i=1}^{L} \delta_{sq}(R_i) \cong \frac{1}{L}\sum_{i=1}^{L} \sigma_i^2 \frac{1}{12} \beta 2^{-2R_i}, \text{ for large } R_i\text{'s}, \tag{6.2}$$

where σ_i^2 is the variance of the ith coefficient. Minimizing this distortion formula over the choice of $R_1,..., R_L$ that average to R, one obtains the optimal bit allocation:

$$R_i = R + \frac{1}{2}\log\frac{\sigma_i^2}{\left(\prod_{i=1}^{L}\sigma_i^2\right)^{\frac{1}{L}}}. \tag{6.3}$$

Notice that the optimal bit allocation depends only on the distribution of coefficient variances. More bits than the average, *R*, are allocated to the coefficients whose variances are greater that the geometric mean of the coefficient variances. On the other hand, fewer bits are allocated to the coefficients whose variances are smaller than the geometric mean. Substituting these rates into the distortion expression, (6.2), gives

$$D_T(R) \cong \frac{\beta}{12}\left(\prod_{i=1}^{L}\sigma_i^2\right)^{\frac{1}{L}} 2^{-2R}. \tag{6.4}$$

6.2.2 Rate Allocation For Noisy Channels

As for noiseless channels, the procedure for finding optimal bit allocations for noisy channels consists of two steps: (1) finding expressions for the optimal distortion of the quantizer applied to each coefficient, and (2) based on these expressions, obtaining bit allocations that minimize the overall distortion with an overall rate constraint.

To accomplish (1), we will use the Hochwald-Zeger high-resolution theory. As mentioned earlier, this theory applies to a model of joint source-channel coding that does not exactly match our model of joint source-channel coding. We first discuss the Hochwald-Zeger model for joint source-channel coding in the context of scalar quantization. As illustrated in Figure 6.1, the quantizer encoder maps the source sample *Y* into an index set *I*, where the index set *I* is ordinarily a collection of consecutive integers from 1 to 2^m (or 0 to 2^m-1). The quantizer encoder is specified by a partition $S = \{S_i : i \in I\}$ of the sample space, where $S_i = \{y : \alpha(y) = i\}$. The quantization index is then mapped to an *m*-bit binary codeword *J* through a permutation mapping, commonly called an index assignment. The channel encoder employs n-m bits of redundancy which are added to the m-bit binary codeword. The channel codeword therefore contains n bits, and the number of channel codeword bits per source samples is n. Given parameters m and n, the scalar quantizer is optimized for a noiseless channel, and the channel code is designed to minimize decoder error probability, assuming that the channel decoder is maximum-likelihood. The decoded output $\hat{J}$ from the channel decoder is reconstructed to $\hat{Y}$ through the inverse index assignment and the quantizer decoder. The cascade of quantizer encoder, index assignment and channel encoder to be considered as a source-channel encoder e. The cascade of channel decoder, inverse index assignment and quantizer decoder is considered to be a source-channel decoder d.

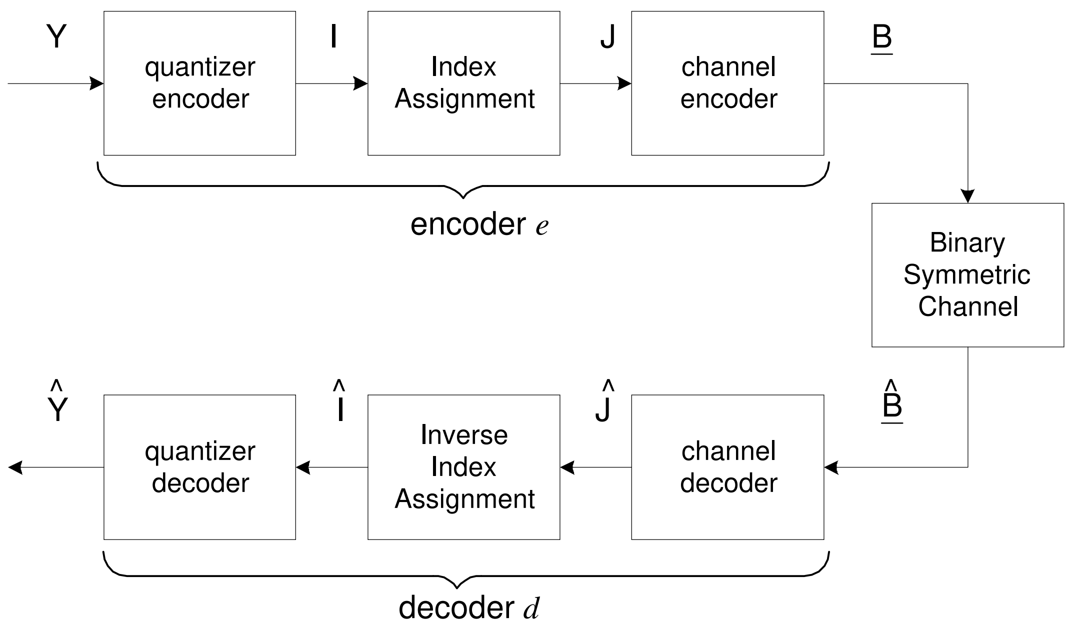


Figure 6.1: Scalar quantizer and channel coder system

Lets us compare this type of code to the channel-optimized quantization in Figure 2.7. Although the index assignment and channel encoder in Figure 6.1 can be considered as the binary encoder in Figure 2.7, the difference is that in Figure 6.1, the quantizer and channel encoder are independently designed (the quantizer is designed for a noiseless channel and the channel code is designed to minimize bit error probability without regard to the source), whereas in Figure 2.7, the quantizer and channel code are jointly designed. On the other hand, the system in Figure 6.1 is like tandem coding except that the input blocklength of the channel code is constrained to match the output blocklength of the source code.

Before describing the model for transform coding based on the new type of joint source-channel coding we review Hochwald and Zeger's asymptotic distortion formula for the case of scalar quantization and mean-squared error. In Figure 6.1, the transmission rate per source sample is $R = n$, the channel code rate is $r = m/n$ and the bit error probability of BSC is ε. Let $D^*(r,R,\varepsilon)$ be the least distortion, averaged over all index assignments, for the system with quantizer rate rR and channel code rate r. They showed that for small bit-error probability ε and large R, the channel code rate $r(R,\varepsilon)$ minimizing $D^*(r,R,\varepsilon)$ over all channel code rates satisfies

$$r_{ex}(R,\varepsilon) \le r(R,\varepsilon) \le r_{sp}(R,\varepsilon) \tag{6.5}$$

where $r_{ex}(R,\varepsilon)$ and $r_{sp}(R,\varepsilon)$ are

$$r_{ex}(R,\varepsilon) = 1 - 4\frac{\log\log(1/\varepsilon)}{\log(1/\varepsilon)} + O(\frac{1}{\log(1/\varepsilon)}) + o(1)\,, \tag{6.6}$$

and

$$r_{sp}(R,\varepsilon) = 1 - 2\frac{\log\log(1/\varepsilon)}{\log(1/\varepsilon)} + O(\frac{1}{\log(1/\varepsilon)}) + o(1)\,. \tag{6.7}$$

The $O(1/\log(1/\varepsilon))$ term goes to zero as ε goes to zero, uniformly in R, and the $o(1)$ term goes to zero as $R \to \infty$. Also for the optimal channel code rate, i.e. $r(R,\varepsilon)$, the corresponding distortion obeys

$$D^*(R,\varepsilon) = 2^{-2Rr+a_R}, \tag{6.8}$$

where $|a_R| \le c$ for some $c > 0$ and all R sufficiently large.

We now want to express the distortion of transform coding over noisy channels using the above asymptotic expression for distortion. The transform coding system is illustrated in Figure 6.2, where the encoder e_i consists of a quantizer encoder, index assignment and channel encoder as in Figure 6.1, and the decoder d_i consists of a channel decoder, inverse index assignment and quantizer decoder, again as in Figure 6.1. An L-dimensional transform T maps L source samples into an L-dimensional vector of coefficients. Each coefficient is then encoded by the encoder e_i and sent over a BSC. We

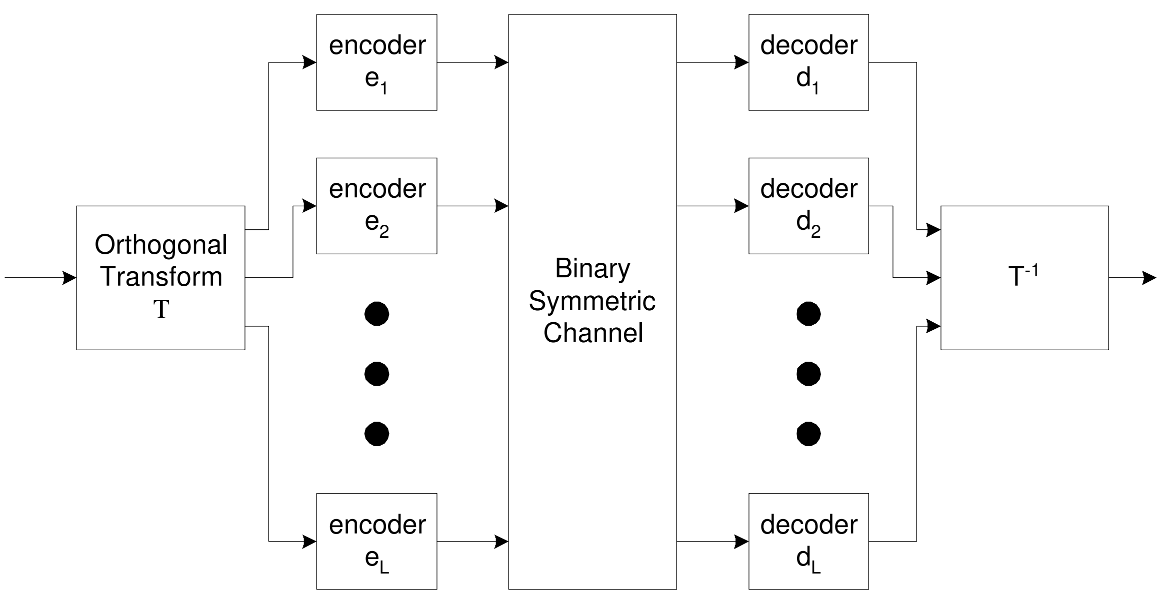


Figure 6.2: The transform coding system for analysis of rate allocation for noisy channels

assume that the source is Gaussian, and so the transform coefficients are Gaussian. Let $D_0(R,r,\varepsilon)$ be the distortion when a Gaussian source with mean zero and unit variance is used in a system consisting of a scalar quantizer with rate rR, which is optimized for noiseless channels, and a channel code with rate r chosen to have smallest possible error probability. Then one can easily show that the distortion for a Gaussian source with variance σ^2 is

$$D_\sigma(R,r,\varepsilon) = \sigma^2 D_0(R,r,\varepsilon) \tag{6.9}$$

and that if $r(R,\varepsilon)$ minimizes $D_0(R,r,\varepsilon)$, then it also minimizes $D_\sigma(R,r,\varepsilon)$. Let $D_0^*(R,\varepsilon)$ and $D_\sigma^*(R,\varepsilon)$ denote the minimum of $D_0(R,r,\varepsilon)$, respectively, $D_\sigma(R,r,\varepsilon)$ over all choices of r and let $r_0(R,\varepsilon)$ and $r_\sigma(R,\varepsilon)$, respectively, denote the minimizing choice of r. Using (6.9) we have

$$D_\sigma^*(R,\varepsilon) = \sigma^2 D_0^*(R,\varepsilon) = \sigma^2 2^{-2Rr(R,\varepsilon)+a_{0,R}}, \tag{6.10}$$

where $a_{0,R}$ is the constant in (6.8) for the unit variance Gaussian source, and $|a_{0,R}| \le c_0$. This is achieved with channel code rate $r(R,\varepsilon)$ satisfying (6.5).

We now return to transform coding. Applying (6.10), if the ith coefficient is allocated rate R_i, then the distortion for the ith coefficient is

$$D_\sigma(R_i,\varepsilon) = \sigma_i^2 2^{-2R_i r(R_i,\varepsilon)+a_{0,R_i}} \tag{6.11}$$

where $r_{ex}(R_i,\varepsilon) \le r(R_i,\varepsilon) \le r_{sp}(R_i,\varepsilon)$ and σ_i^2 is the variance of the ith coefficient. We assume that $\sigma_1^2 \ge \sigma_2^2 \ge \cdots \ge \sigma_L^2$. Since the distortion of the transform code for a given bit allocation is the average of each coefficient's distortion, the distortion is

$$D_T(R_1,\cdots,R_L,\varepsilon) = \frac{1}{L}\sum_{i=1}^{L} D_{\sigma_i}(R_i,\varepsilon) = \frac{1}{L}\sum_{i=1}^{L} \sigma_i^2 2^{-2R_i r(R_i,\varepsilon)+a_{0,R_i}}. \tag{6.12}$$

Applying (6.5), we have

$$\frac{1}{L}\sum_{i=1}^{L}\sigma_i^2 2^{-2R_i r_{sp}(R_i,\varepsilon)a_{0,R_i}} \le D_T(R_1,\cdots,R_L,\varepsilon) \le \frac{1}{L}\sum_{i=1}^{L}\sigma_i^2 2^{-2R_i r_{ex}(R_i,\varepsilon)+a_{0,R_i}} \tag{6.13}$$

Given some rate *R*, we wish to find $R_1,\cdots,R_L$ that average to *R* and minimize $D_T(R_1,\cdots,R_L,\varepsilon)$. We now assume ε is so small and *R* is so large that the R_i's which minimize $D_T(R_1,\cdots,R_L,\varepsilon)$ are sufficiently large that we may use the approximation

$$r_{ex}(R_i,\varepsilon) = r'_{ex}(\varepsilon) \tag{6.14}$$

and

$$r_{sp}(R_i,\varepsilon) = r'_{sp}(\varepsilon), \tag{6.15}$$

where

$$r'_{ex}(\varepsilon) = 1 - 4\frac{\log\log(1/\varepsilon)}{\log(1/\varepsilon)} \tag{6.16}$$

and

$$r'_{sp}(\varepsilon) = 1 - 2\frac{\log\log(1/\varepsilon)}{\log(1/\varepsilon)}. \tag{6.17}$$

Then, we have

$$\min_{\frac{1}{L}\sum_{i=1}^{L}R_i \le R}\left(\frac{1}{L}\sum_{i=1}^{L}\sigma_i^2 2^{-2R_i r'_{sp}(\varepsilon)+a_{0,R_i}}\right) \le D_{tr}(R,\varepsilon) \le \min_{\frac{1}{L}\sum_{i=1}^{L}R_i \le R}\left(\frac{1}{L}\sum_{i=1}^{L}\sigma_i^2 2^{-2R_i r'_{ex}(\varepsilon)+a_{0,R_i}}\right). \tag{6.18}$$

Since the right and left parts differ only in terms of r'_{sp} and r'_{ex}, we focus on minimizing the right side. We denote the function being minimized on the right side as $D_{ub}(R_1,\cdots,R_L,\varepsilon)$. In minimizing $D_{ub}(R_1,\cdots,R_L,\varepsilon)$, we observe that when *R* is large, the R_i's will be large. Therefore, the exponents in $D_{ub}(R_1,\cdots,R_L,\varepsilon)$ will be dominated by the terms $-2R_i r'_{ex}(\varepsilon)$. As a result, when minimizing $D_{ub}(R_1,\cdots,R_L,\varepsilon)$, we ignore the a_{0,R_i} term. We can then write the minimization of $D_{ub}(R,\varepsilon)$ with a given average rate *R* using Lagrange multiplier as the minimization of

$$J = D_{ub}(R,\varepsilon) - \lambda\left(R - \frac{1}{L}\sum_{i=1}^{L}R_i\right). \tag{6.19}$$

Differentiating (6.19) with respect to R_i, we have

$$\frac{\partial J}{\partial R_i} = \frac{\partial D_{ub}}{\partial R_i} + \lambda \frac{1}{L} \cong \frac{1}{L} 2r'_{ex} \sigma_i^2 \ln 2 \; 2^{-2r'_{ex}(\varepsilon)R_i} + \lambda \frac{1}{L}. \tag{6.20}$$

Setting the derivative (6.20) to 0, we obtain

$$R_i = \frac{1}{2r'_{ex}(\varepsilon)} \log_2 \frac{2r'_{ex}(\varepsilon)\sigma_i^2 \ln 2}{\lambda}, \text{for } i = 1,2,\cdots,L. \tag{6.21}$$

Substituting this in the constraint, 1/L $\sum_{i=1}^{L} R_i$ = R, to find λ, we have

$$\frac{1}{L}\sum_{i=1}^{L} R_i = \frac{1}{L}\sum_{i=1}^{L} \frac{1}{2r'_{ex}(\varepsilon)} \log_2 \frac{2r'_{ex}(\varepsilon)\sigma_i^2 \ln 2}{\lambda} = R \tag{6.22}$$

Equation (6.22) gives

$$\lambda = \frac{2^{-2r'_{ex}(\varepsilon)R}}{2r'_{ex}(\varepsilon)\ln 2\left(\prod_{i=1}^{L}\sigma_i^2\right)^{1/L}}. \tag{6.23}$$

Inserting (6.23) into (6.21), we have the optimal bit allocation:

$$R_i = R + \frac{1}{2r'_{ex}(\varepsilon)} \log_2 \frac{\sigma_i^2}{\left(\prod_{i=1}^{L}\sigma_i^2\right)^{\frac{1}{L}}}. \tag{6.24}$$

Substituting these rates into the distortion expression gives the minimum distortion using the optimal bit allocation:

$$D_{ub}(R,\varepsilon) = \frac{1}{L}\sum_{i=1}^{L}\left(\prod_{j=1}^{L}\sigma_j^2\right)^{\frac{1}{L}} 2^{-2Rr'_{ex}(\varepsilon)+b_R} = \left(\prod_{j=1}^{L}\sigma_j^2\right)^{\frac{1}{L}} 2^{-2Rr'_{ex}(\varepsilon)+b_R}, \tag{6.25}$$

where b_R, $|b_R| \le c_0$, is a term that accounts for the a_{R_i}'s that we ignored.

The same expression for the optimal bit allocation given by (6.24) can be obtained using the following fact:

Let $g_1(R), g_2(R), \cdots, g_L(R)$ be differentiable, nonincreasing functions and let $R_1, R_2, \cdots, R_L$ minimize $\frac{1}{L}\sum_{i=1}^{L} g_i(R_i)$ subject to the constraint that $\frac{1}{L}\sum_{i=1}^{L} R_i = R$. Then

$$\frac{d}{dR_i} g_i(R_i) = \frac{d}{dR_j} g_j(R_j), \quad \text{for all } i, j\,. \tag{6.26}$$

That is, for the optimum choice of $R_1, R_2, \cdots, R_L$, the slopes of the functions are the same.

Applying this fact to the bit allocation problem, we have a constant g such that for optimum choice of $R_1, R_2, \cdots, R_L$,

$$g = \frac{d}{dR_i} \sigma_i^2 2^{-2R_i r'_{ex}(\varepsilon)} \cong \sigma_i^2 2^{-2R_i r'_{ex}(\varepsilon)} (-2r'_{ex}(\varepsilon) \ln 2), \text{ for all } i\,. \tag{6.27}$$

Solving (6.27) yields

$$R_i = \frac{1}{2r'_{ex}(\varepsilon)} \log_2 \sigma_i^2 - \frac{1}{2r'_{ex}(\varepsilon)} \log_2 g\,. \tag{6.28}$$

Since the R_i's must satisfy the average rate constraint, inserting (6.28) into the constraint, 1/L $\sum_{i=1}^{L} R_i = R$, gives

$$-\frac{1}{2r'_{ex}(\varepsilon)} \log_2 g = R - \frac{1}{L}\sum_{i=1}^{L} \frac{1}{2r'_{ex}(\varepsilon)} \log_2 \sigma_i^2 = R - \frac{1}{2r'_{ex}(\varepsilon)} \log_2 \left(\prod_{i=1}^{L} \sigma_i^2\right)^{\frac{1}{L}}. \tag{6.29}$$

Combining (6.28) and (6.29), we get the optimal bit allocation:

$$R_i = R + \frac{1}{2r'_{ex}(\varepsilon)} \log_2 \frac{\sigma_i^2}{\left(\prod_{i=1}^{L} \sigma_i^2\right)^{\frac{1}{L}}}. \tag{6.30}$$

Using the resulting distortion expressed by (6.25) and a similar expression for $D_{lb}(R_1,\cdots,R_L,\varepsilon)$, the minimum distortion of transform codes, given by (6.18), is

$$\left(\prod_{i=1}^{L}\sigma_i^2\right)^{\frac{1}{L}} 2^{-2Rr'_{sp}(\varepsilon)+b_R} \le D_T(R,\varepsilon) \le \left(\prod_{i=1}^{L}\sigma_i^2\right)^{\frac{1}{L}} 2^{-2Rr'_{ex}(\varepsilon)+b_R}, \qquad (6.31)$$

where the bit allocation for the lower bound is

$$R_i^l = R + \frac{1}{2r'_{sp}(\varepsilon)}\log_2 \frac{\sigma_i^2}{\left(\prod_{i=1}^{L}\sigma_i^2\right)^{\frac{1}{L}}} \qquad (6.32)$$

and the bit allocation for the upper bound is

$$R_i^u = R + \frac{1}{2r'_{ex}(\varepsilon)}\log_2 \frac{\sigma_i^2}{\left(\prod_{i=1}^{L}\sigma_i^2\right)^{\frac{1}{L}}}. \qquad (6.33)$$

Notice that the upper and lower bounds in (6.31) are quite close to each other. Thus, for some $r^*(\varepsilon)$ between $r'_{ex}(\varepsilon)$ and $r'_{sp}(\varepsilon)$, the distortion can be expressed by

$$D_T(R,\varepsilon) = \left(\prod_{i=1}^{L}\sigma_i^2\right)^{\frac{1}{L}} 2^{-2Rr^*(\varepsilon)+b_R}. \qquad (6.34)$$

and this distortion can be achieved by the following bit allocation

$$R_i = R + \frac{1}{2r^*(\varepsilon)}\log_2 \frac{\sigma_i^2}{\left(\prod_{i=1}^{L}\sigma_i^2\right)^{\frac{1}{L}}}. \qquad (6.35)$$

This distortion expression (6.34) is similar to that for noiseless channels, (6.4), except that the $Rr^*(\varepsilon)$ factor replaces R. Compared with the case of noiseless channels, the optimal bit allocation for noisy channels, given by (6.35), has $1/(2r^*(\varepsilon))$ factor instead of 1/2 in the expression for noiseless channels, given by (6.3). Since $0 < r^*(\varepsilon) \le 1$, we have

$1/(2r^*(\varepsilon)) \geq 1/2$. Thus, even more bits need to be allocated to the coefficients whose variances are greater than the geometric mean of the coefficient variances, compared to the bit allocation for noiseless channel. On the other hand, even fewer bits are to be allocated to the coefficients whose variances are smaller than the geometric mean. Since $r'_{ex}(\varepsilon) \leq r^*(\varepsilon) \leq r'_{sp}(\varepsilon)$, the optimal bit allocation given by (6.35) lies between the bit allocations for the upper and lower bounds, given by (6.32) and (6.33).

Figure 6.3 shows the optimal bit allocation for transform codes over noiseless channels and the bit allocation bounds for transform codes over noisy channels for $\varepsilon = 10^{-5}$ and $\varepsilon = 10^{-2}$. As shown in the figure, even more bits are allocated to the coefficients whose variances are greater than the geometric mean of the coefficient variances, compared to the bit allocation for noiseless channel. Furthermore, as the channel becomes noisier, the amount of bits allocated to the coefficients whose variances are greater than the geometric mean increases.

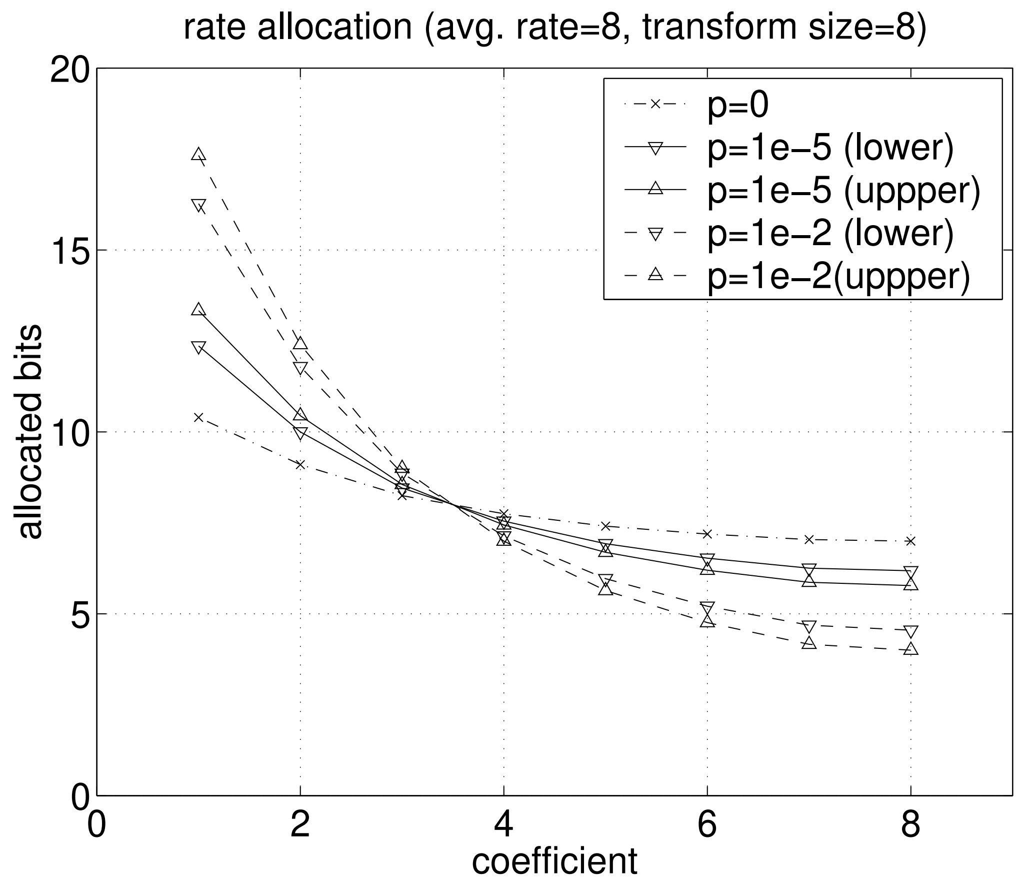


Figure 6.3. The bit allocation for ideal channel and noisy channel ($\varepsilon=10^{-5}$ and $\varepsilon=10^{-2}$)

6.3 Experimental Results

This section presents the numerical results of the optimal bit allocation for the channel-optimized transform coding system. As the data source, we choose a one-dimensional, stationary Gauss-Markov random sequence with zero mean, unit variance, and correlation coefficient ρ. The Karunen-Loeve transform is used. The bit allocation of channel-optimized transform code is performed based on the method of Vaishampayan and Farvardin's [5]. The allowable bits to each coefficient are limited to 12.

Figure 6.4 and 6.5 shows the resulting bit allocations for channel-optimized transform coding for $p = 10^{-5}$ and $p = 10^{-2}$, respectively. The transform size is 8 and the average rate R is 8. In these figures, the dashed line marked with 'o's shows the bit allocation for noiseless channels. The dash-dot line (marked with 'x's) shows the rate allocation obtained by the theoretical expression given (6.3) for noiseless channels. We can see that the bit allocation obtained by numerical algorithm fits well into the theoretical analysis. The small discrepancy between the two values comes from the fact that the theoretical formula gives non-integer rates, whereas the numerical analysis uses only integer values for the quantizer rates.

For the noisy channels, the theoretical bounds for bit allocation given by (6.32) and (6.33) is shown by the dashed line and solid line without any markers. The solid line with '*'s shows the resulting bit allocation using numerical algorithm. We can see that the bit allocation using the numerical algorithm follows the general trend predicted by theoretical analysis. As shown in these figures, for the noisy channel, even more bits are allocated to the coefficients whose variances are greater than the geometric mean of the coefficient variances, compared to the bit allocation for noiseless channel. On the other hand, fewer bits are allocated to the coefficients whose variances are smaller than the geometric mean. Figure 6.5 is the same as Figure 6.4 except the channel has $p = 10^{-2}$. Notice that the bits allocated to the first and second coefficient are same. This is because the allowable bits are limited to 12. If we increase the maximum bit allocation, we can expect that the bits allocated to the first coefficient will be greater than those allocated to the second coefficient. Comparing the bit allocation for the two noisy channels, we can see that as the channel gets noisier, the difference between the bits allocated for the noisy channel and the bits allocated for the noiseless channel gets bigger. For example, for noiseless channel, 9 bits are allocated to the second coefficient, for the channel with $p = 10^{-5}$, 10 bits are allocated, whereas 12 bits are allocated for the channel with $p = 10^{-2}$.

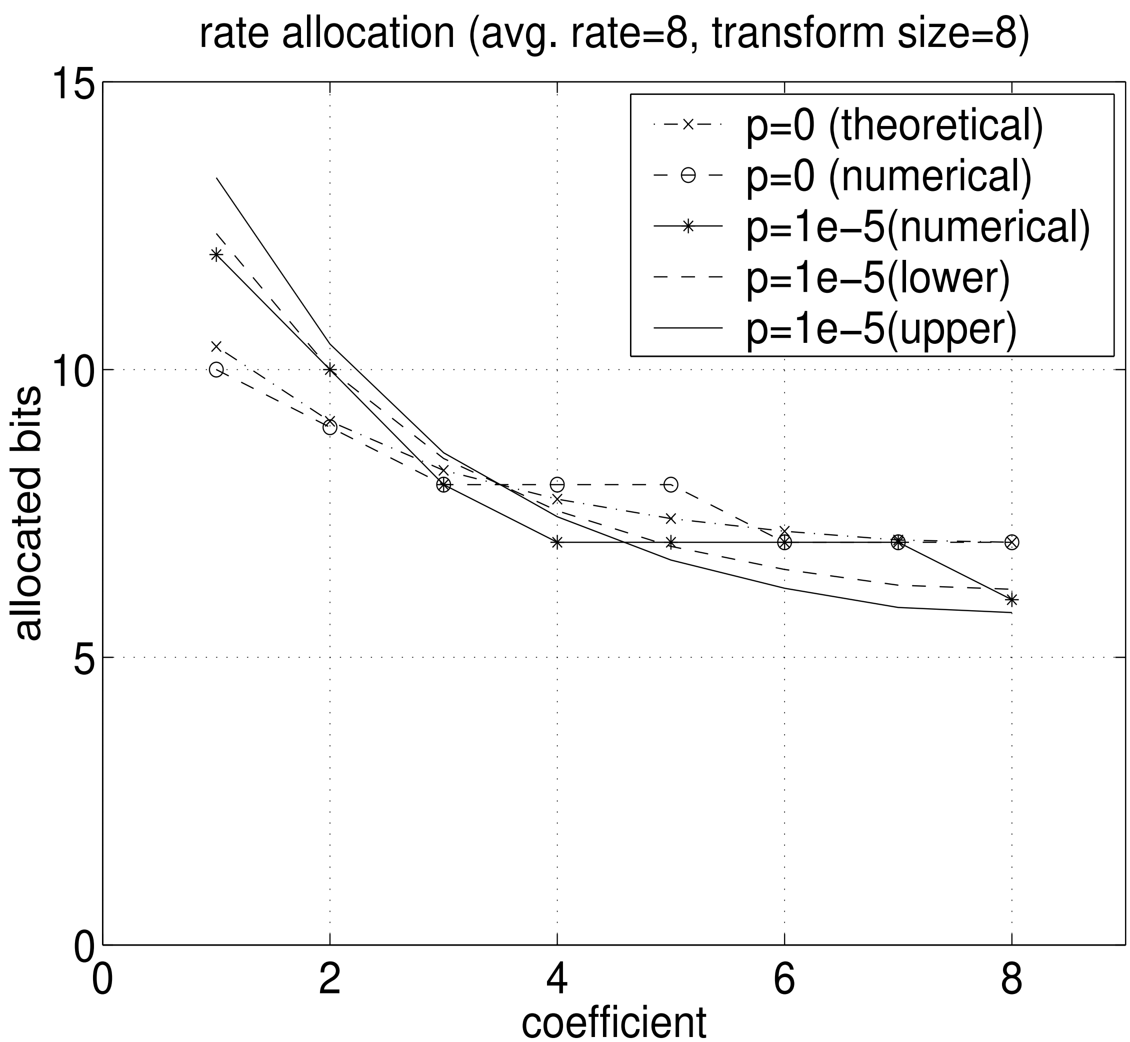


Figure 6.4. The bit allocation of channel-optimized transform codes
(KLT, ρ=0.9 and p=10^{-5})

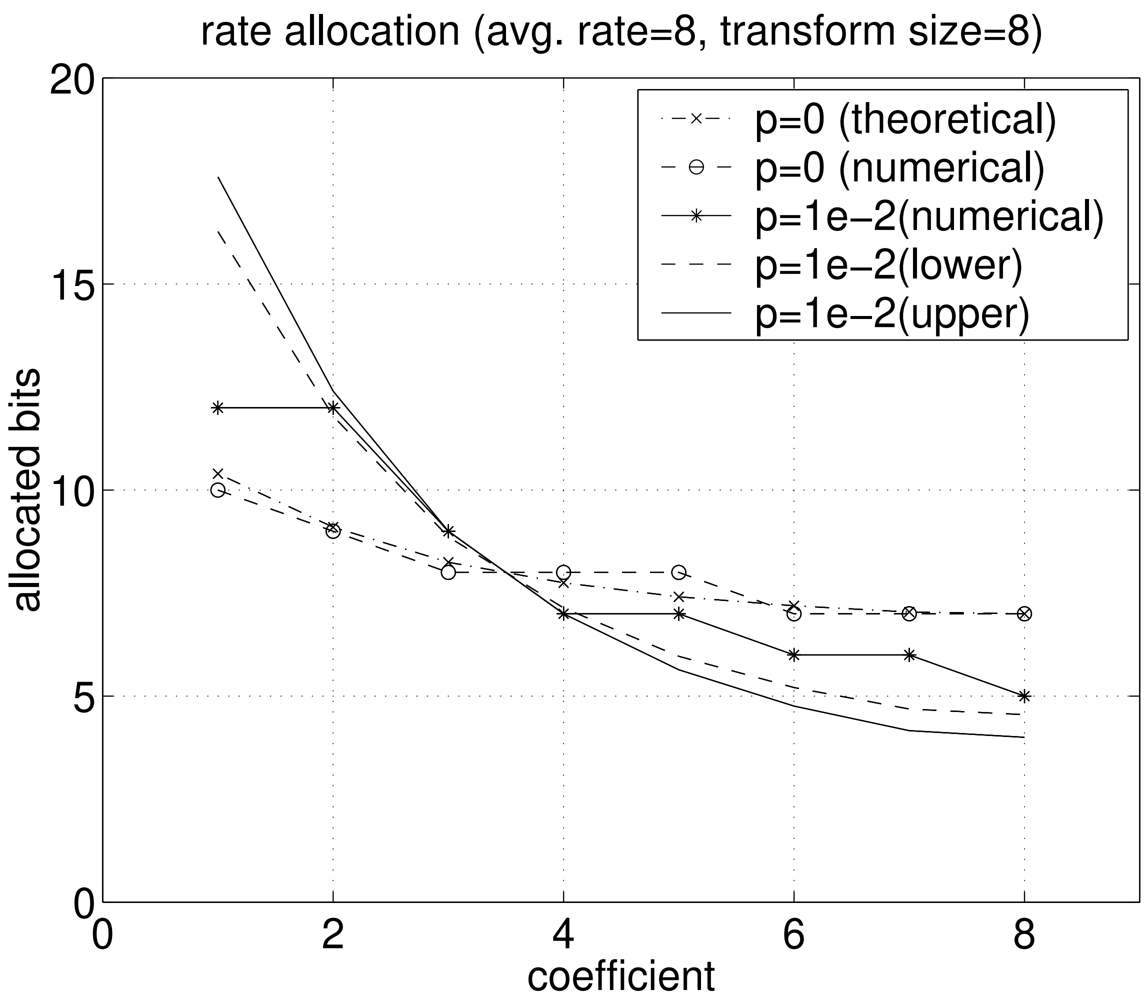


Figure 6.5. The bit allocation of channel-optimized transform codes
(KLT, ρ=0.9 and $p=10^{-2}$)

CHAPTER VII
CONCLUSIONS AND FUTURE WORK

7.1 Summary

In this thesis, we have compared two common source-channel coding strategies, joint and tandem source-channel coding, on the basis of distortion vs. complexity and distortion vs. delay. We have analyzed specific representatives of each when transmitting analog data samples across a binary symmetric channel. Channel-optimized transform coding is the joint source-channel strategy; transform coding with Reed-Solomon coding is the tandem strategy. We derived formulas for the mean-squared error, computational complexity and delay for each strategy, and we optimized each system so as to minimize distortion subject to a complexity or delay constraint, for data modeled as Gauss-Markov. The results of such optimizations suggest there exists a complexity threshold such that when the number of available operations per data sample is above this threshold, tandem coding is better, and when below the threshold, channel-optimized transform coding is better. Like the results for complexity constraints, the optimization results with delay constraints suggest there is a delay threshold such that tandem coding is better than joint when and only when the permissible delay is above this threshold.

Our investigation was also expanded to unequal error protection in tandem source-channel coding, where the transform coefficients of the transform coders have different levels of importance. We have applied the unequal error protection scheme to the tandem system and investigated the performance of unequal error protection system with a complexity constraint. The unequal error protection scheme showed improved performance under a given complexity constraint. Thus it reduces the complexity threshold relative to that of an equal error protection system.

We have also derived an asymptotic expression for the optimal rate allocation for noisy channels using high-resolution analysis. The resulting expression suggests that even more bits need to be allocated to the coefficients whose variances are greater than the geometric mean of such variances, compared to the bit allocation for noiseless channels. On the other hand, fewer bits are to be allocated to the coefficients whose variances are smaller than the geometric mean.

7.2 Future Work

There remains the need for much theoretical work to clarify the performance of these types of systems. One of the problems is to investigate the relation between the distortion due to the source codes and the distortion due to the channel code and the channel errors. Especially, under a complexity constraint, the distortion can be divided as following:

$$D = D_S(R_S, C_S) + D_C(R_C, C_C), \tag{7.1}$$

where C_S (C_C) is the complexity allocated into source (channel) coding. It would be interesting to develop a theory to describe the behavior of these distortion functions under transmission and complexity constraints. Another problem is to design a more sophisticated algorithm for allocating rate and complexity in the unequal error protection system. The more sophisticated algorithm might improve the performance of the unequal error protection systems and make the threshold even smaller.

Most joint source-channel codes have been designed for a particular source and channel. However, modern communication networks are characterized by their variability. For example, in wireless communication links, the characteristics of the channel are continually varying. So, systems that are robust to changes in channel characteristics are needed.

There has been much interest in fading channels for mobile multi-user communication and packet channels for Internet communication. Since Shannon's source and channel separation theory is based on point-to-point communication, and thus may not be applicable for these channels. Joint design of source and channel code may lead to much improved performance.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, pp.379-423, July 1948.

[2] A.J. Kurtenbach and P.A. Wintz, "Quantizing for Noisy Channels," *IEEE Trans. Comm. Technol.,* vol. COM-17, pp. 291-302, Apr. 1969.

[3] H. Kumazawa, M. Kasahara, and T. Namekawa, " A construction of vector quantizers for noisy channels," *Electron. And Eng. Japan,* vol. 67-B, pp. 39-47, 1984, translated from *Denshi Tsushin Gakkai Ronbunshi,* vol. 67-B,pp. 1-8, Jan. 1984.

[4] N. Favardin and V. Vaishampayan, "Optimal Quantizer Design for Noisy Channels" An Aproach to Combined Source-Channel Coding", *IEEE Transaction on Information Theory,* vol. IT-33, No. 6, pp827-838, Nov. 1987.

[5] V. Vaishampayan and N. Favardin, "Optimal Block Cosine Transform Image Coding for Noisy Channels," *IEEE Transasction on Communications.* vol. COM-38, No. 3, pp. 327-336, Mar. 1990.

[6] N. Favardin and V. Vaishampayan, "On the Performance and Complexity of Channel-Optimized Vector Quantizers," *IEEE Trans. on Information Theory*, vol. IT-37, No.1, pp 155-160, Jan. 1991

[7] N. Rydbeck and C.-E. W. Sundberg, "Analysis of digital errors in nonlinear PCM systems," *IEEE Trans. Commun.*, vol. COM-24, pp 59-65, Jan. 1976

[8] J.R.B. DeMarca and N. S. Jayant, "An algorithm for assigning binary indices to the codevectorts of multidimensional quantizers," in *Proc. IEEE Int. Conf. Commnunications*, June 1987, pp.1128-1132.

[9] K. Zeger and A. Gersho, "Pseudo-Gray coding," *IEEE Trans. Commun.*, vol. 38, pp.2147-2156, May 1990.

[10] T.R. Crimmins, H. M. Horwitz, C.J. Palermo, and R.V. Palermo, "Minimization of mean-squared error for data transmitted via group codes," *IEEE Trans. Inform. Theory*, vol. IT-15, pp.72-78, Jan. 1969.

[11] S. W. McLaughin, D. L. Neuhoff, and J. K. Ashley, "Optimal binary index assignments for a class of equiprobable scalar and vector quantizers," *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 2031-2037, Nov. 1995

[12] A. Mehes and K. Zeger, "Binary Lattice Vector Quantization With Linear Block Codes and Affine Index Assignments," *IEEE Trans. Inform. Theory*, vol. 44, pp. 79-94, 1998.

[13] J. W. Modestino and D. G. Daut, "Combined source-channel coding of images," *IEEE Trans. Commun.*, vol. COM-27, pp. 1644-1659, Nov. 1979

[14] J. W. Modestino ,D. G. Daut and A. L. Vickers, " Combined source-channel coding of images using the block cosine transform," *IEEE Trans. Commun.*, vol. COM-29, pp. 1261-1274, 1981

[15] J. Hagenauer, "Rate-Compatible Punctured Convolutional Codes(RCPC Codes) and their Applications," J. W. Modestino and D. G. Daut, "Combined source-channel coding of images," *IEEE Trans. Commun.*, vol. COM-36, pp. 389-400, 1988

[16] N. Tanabe and N. Farvardin, "Subband Image coding using entropy-coded quantization over noisy channels," J. W. Modestino and D. G. Daut, "Combined source-channel coding of images," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 926-943, June 1992.

[17] R.V. Cox, J. Hagenauer, N. Seshadri, and C.-E. W. Sundberg, " Variable rate sub-band speech coding and matched convolutional channel coding for mobile radio channels," *IEEE Trans. Signal Processing*, vol. 39, pp. 1717-1731, Aug. 1991.

[18] P. G. Sherwood and K. Zeger, "Progressive image coding on noisy channels," *Proc. IEEE Data Compress. Conf.*, Mar. 1997, pp. 72-81.

[19] P. G. Sherwood and K. Zeger, "Progressive image coding on noisy channels," *IEEE Signal Processing Letters*, vol. 4, No. 7, July 1997.

[20] A. J. Goldsmith and M. Effros, "Joint Design of Fixed-Rate Source codes and Multiresolution Channel Codes," *IEEE Trans. Commun.*, vol. 46, pp. 1301-1312, 1998.

[21] E. Ayanoglu and R. M. Gray, " The design of joint source and channel trellis waveform coders," *IEEE Trans. Inform. Theory,* vol. IT-33,pp.855-865, Nov. 1987.

[22] T. R. Fischer and M. W. Marcellin, "Joint trellis coded quantization/modulation," *IEEE Trans. Commun.*, vol. 39, pp. 172-176, Feb. 1991.

[23] M. Wang and R. R. Fischer, "Trellis-coded quantization designed for noisy channel," *IEEE Trans. Inform. Theory,* vol. IT-40, pp.1797-1802, Nov. 1994.

[24] C. A. Vaishampayan and N. Farvardin, "Joint design of block source codes and modulation signal sets," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1230-1248, July 1992.

[25] K.-P. Ho and J. M. Kahn, "Combined source-channel coding using channel-optimized quantizer and multicarrier modulation," in *Proc. IEEE ICC'96*, June 1996, pp. 1323-1327.

[26] B. Hochwald and K. Zeger, "Tradeoff Between Source and Channel coding," *IEEE Trans. Inform. Theory*, vol. 43, pp.1412-1424, Sep. 1997.

[27] B. Hochwald, "Tradeoff Between Source and Channel coding on a Gaussian Channel," *IEEE Trans. Inform. Theory*, vol. 44, pp.3044-3055, Nov. 1998.

[28] N. S. Jayant & P. Noll, Digital Coding of Waveforms, Prentice-Hall,1984.

[29] R.J. McEliece and L. Swanson, "On the decoder error probability for Reed-Solomon codes," *IEEE Trans. Inform. Theory,* vol. 32, pp.701-703, Sep. 1986.

[30] M. Vetterli, H.J. Nussbaumer, "Simple FFT and Algorithms with Reduced Number of Operations," Signal Processing, Vol. 6, pp. 267-278, 1984.

[31] C. W. Kok, "Fast Algorithm for Computing Discrete Cosine Transform," IEEE Tran. on Signal processing, Vol. 45, No. 3, pp. 757-760, March, 1997.

[32] A. Gray and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer academic publishers, 1991.

[33] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Inform. Theory,* pp.7-12, March 1960.

[34] P.F. Panter and W. Dite, "Quantization distortion in pulse count modulation with nonuniform spacing of levels," *Proc. IRE*, Vol. 39, pp. 44-48, Jan. 1951.

[35] J.J.Y. Huang and P.M. Schulthesis, "Block quantization of correlated Gaussian variable," *IEEE Trans. Commun.*, Vol. COM-11, pp. 289-296, Sept. 1963.

[36] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, Vol. 44 , No. 6, October, 1998.