# Dynamic Information and Constraints in Source and Channel Coding

by

Emin Martinian

B.S., University of California at Berkeley (1997)
S.M., Massachusetts Institute of Technology (2000)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2004

Author . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Department of Electrical Engineering and Computer Science
September 1, 2004

Certified by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Gregory W. Wornell
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

## Abstract

This thesis explore dynamics in source coding and channel coding.

We begin by introducing the idea of distortion side information, which does not directly depend on the source but instead affects the distortion measure. Such distortion side information is not only useful at the encoder but under certain conditions knowing it at the encoder is optimal and knowing it at the decoder is useless. Thus distortion side information is a natural complement to Wyner-Ziv side information and may be useful in exploiting properties of the human perceptual system as well as in sensor or control applications. In addition to developing the theoretical limits of source coding with distortion side information, we also construct practical quantizers based on lattices and codes on graphs. Our use of codes on graphs is also of independent interest since it highlights some issues in translating the success of turbo and LDPC codes into the realm of source coding.

Finally, to explore the dynamics of side information correlated with the source, we consider fixed lag side information at the decoder. We focus on the special case of perfect side information with unit lag corresponding to source coding with feedforward (the dual of channel coding with feedback). Using duality, we develop a linear complexity algorithm which exploits the feedforward information to achieve the rate-distortion bound.

The second part of the thesis focuses on channel dynamics in communication by introducing a new system model to study delay in streaming applications. We first consider an adversarial channel model where at any time the channel may suffer a burst of degraded performance (e.g., due to signal fading, interference, or congestion) and prove a coding theorem for the minimum decoding delay required to recover from such a burst. Our coding theorem illustrates the relationship between the structure of a code, the dynamics of the channel, and the resulting decoding delay. We also consider more general channel dynamics. Specifically, we prove a coding theorem establishing that, for certain collections of channel ensembles, delay-universal codes exist that simultaneously achieve the best delay for any channel in the collection. Practical constructions with low encoding and decoding complexity are described for both cases.

Finally, we also consider architectures consisting of both source and channel coding which deal with channel dynamics by spreading information over space, frequency, multiple antennas, or alternate transmission paths in a network to avoid coding delays. Specifically, we explore whether the inherent diversity in such parallel channels should be exploited at the application layer via multiple description source coding, at the physical layer via parallel channel coding, or through some combination of joint source-channel coding. For on-off channel models application layer diversity architectures achieve better performance while for channels with a continuous range of reception quality (e.g., additive Gaussian noise channels with Rayleigh fading), the reverse is true. Joint source-channel coding achieves the best of both by performing as well as application layer diversity for on-off channels and as well as physical layer diversity for continuous channels.

Supervisor: Gregory W. Wornell
Title: Professor of Electrical Engineering and Computer Science

## Acknowledgments

Completing this thesis reminds how much I have learned from my friends, family, and colleagues and the debt of gratitude I owe them.

The most important reason for my success at MIT is my advisor, Gregory Wornell. He taught me how to think about research, how to see the big picture without being distracted by technical details, and how not to give up until such details are resolved. I am also deeply grateful for Greg's wisdom, patience, and friendship. Graduate school contains periods where every idea seems to fail and it is easy to become discouraged. Greg provided encouragement during these times while also explaining how they are essential in learning to think independently.

I would also like to thank Greg and Prof. Alan Oppenheim for bringing together such a wonderful collection of people in the Signals, Information, and Algorithms Laboratory and the Digital Signal Processing Group. I learned a lot from everyone in the group especially Richard Barron, Albert Chan, Brian Chen, Aaron Cohen, Vijay Divi, Stark Draper, Yonina Eldar, Uri Erez, Everest Huang, Ashish Khisti, Nicholas Laneman, Mike Lopez, Andrew Russel, Maya Said, Charles Sestok, Charles Swannack, Huan Yao, and Ram Zamir. Nick and Stark in particular were mentors to me and I fondly remember the many conversations, basketball games, and conferences we enjoyed together. Finally, no student scrambling to submit a paper or finish a thesis can fail to appreciate the value of an excellent system administrator and administrative assistant like Giovanni Aliberti and Tricia Mulcahy.

The members of my thesis committee, Prof. Dave Forney, Prof. Ram Zamir, and Prof. Lizhong Zheng deserve special thanks for their efforts in reading the thesis and offering their advice. During his sabbatical at MIT, Prof. Zamir introduced me to high resolution rate-distortion theory and inspired many of the ideas in Chapters 2 and 3. Also, Prof. Forney's comments on Chapter 4 and Prof. Zheng's remarks about possible applications of the ideas in Chapter 3 to channel coding were especially helpful.

Various internships were an important complement to my time at MIT. Working with Carl-Erik Sundberg at Lucent Technologies' Bell Labs was valuable in understanding research problems from an industrial perspective. Working at Bell Labs sparked my interest in low delay coding and led to Chapters 6–9. While at Bell Labs, I also had the pleasure to meet Ramesh Johari who has been a lunch companion and friend ever since. I am grateful to Ramesh for reminding me how much I enjoyed MIT on the (few) occasions I complained about graduate school and for many other interesting observations.

Visits to Hewlett-Packard Labs sponsored by the HP-MIT Alliance were also a stimulating and memorable experience. In particular, an argument between Nick, John Apostolopoulos, Susie Wee, and I during one such visit led to the research described in Chapter 10. Mitch Trott's wry advice, culinary knowledge, and thoughts about research were another highlight of visiting HP and Chapter 8 is one result of our collaboration.

Jonathan Yedidia hired me for a wonderful summer at Mitsubishi Electronic Research Labs and helped me see electrical engineering from a fresh perspective. I am especially grateful to Jonathan for teaching me about belief propagation and taking a chance on my strange idea about source coding with codes on graphs described in Chapter 4.

Finally, I would like to thank my family and friends. My parents, my aunt and uncle, Monica and Reuben Toumani, and my second family, the Malkins, provided much appreciated love, encouragement and support. Alex Aravanis, Jarrod Chapman, and Mike Malkin have my gratitude for their warm friendship and good company. The conversations we had about science, math and engineering over the years are what inspired me to pursue graduate

studies in engineering instead of seeking my fortune in another career. In Boston I would like to thank Orissa, Putney, and Adelle Lawrence, and Joyce Baker for helping me adjust to life here.

Most of all I would like to thank my wife Esme. Her love, sense of humor, and friendship have made these past six years in Boston the best time of my life. If not for her, I would never have come to MIT and never completed this thesis.

*This research was supported by the National Science Foundation through grants and a National Science Foundation Graduate Fellowship and by the HP-MIT Alliance.*

*To my parents Medea and Omega,*
*and their efforts for my education*

## Bibliographic Note

Portions of Chapter 2 were presented at the Data Compression Conference [124] and the International Symposium on Information Theory [123] with co-authors Gregory W. Wornell and Ram Zamir.

Chapter 4 is joint work with Jonathan S. Yedidia and was presented at the Allerton conference [125].

Chapter 5 is joint work with Gregory W. Wornell and will be presented at the forty-second annual Allerton conference in Monticello, IL in October 2004.

Portions of Chapter 9 were presented at the Allerton conference [122] with co-author Gregory W. Wornell.

Chapter 10 is joint work with J. Nicholas Laneman, Gregory W. Wornell, and John G. Apostolopoulos and has been submitted to the Transactions on Information Theory as well as presented in part (with Susie Wee as additional co-author) at the International Conference on Communications [96] and the International Symposium on Information Theory [95].

# Contents

# List of Figures

# List of Tables

Chapter 1

# Introduction

Communication channels, data sources, and distortion measures that vary during compression or communication of the data stream are the focus of this thesis. Specifically, our goal is to understand how such model dynamics affect the fundamental limits of communication and compression and to understand how to design efficient systems. The key differences between static and dynamic scenarios are knowledge and delay: learning the system state and averaging over a long enough period essentially makes these two scenarios equivalent. Thus our main questions can be summarized as "What are the effects of distributed, imperfect, or missing knowledge of state information?" and "What are the effects of delay constraints?". After outlining our approach to these questions in the rest of this chapter, we explore the former mostly in the context of source coding in Part I and the latter mostly in the context of channel coding in Part II.

## ■ 1.1 Distributed Information in Source Coding

### ■ 1.1.1 Distortion Side Information

In the classical source coding problem, an encoder is given a source signal, $\mathbf{s}$, and must represent it using $R$ bits per sample. Performance is measured by a distortion function, $d(s, \hat{s})$, which describes the cost of quantizing a source sample with the value $s$ to $\hat{s}$. In many applications, however, the appropriate distortion function depends on a notion of quality which varies dynamically and no static distortion model is appropriate.

For example, a sensor may obtain measurements of a signal along with some estimate of the quality of such measurements. This may occur if the sensor can calibrate its accuracy to changing conditions (*e.g.*, the amount of light, background noise, or other interference present), if the sensor averages data for a variety of measurements (*e.g.*, combining results from a number of sub-sensors) or if some external signal indicates important events (*e.g.*, an accelerometer indicating movement). Clearly the distortion or cost for coding a sample should depend on the dynamically changing reliability.

Perceptual coding is another example of a scenario with dynamically varying notion of distortion. Specifically, certain components of an audio or video signal may be more or less sensitive to distortion due to masking effects or context [85]. For example errors in audio samples following a loud sound, or errors in pixels spatially or temporally near bright spots or motion are perceptually less relevant. Similarly, accurately preserving certain edges or textures in an image or human voices in audio may be more important than preserving background patterns/sounds.

In order to model the dynamics of the distortion measure, we introduce the idea of distortion side information. Specifically, we model the source coding problem as mapping a source, $\mathbf{s}$, to a quantized representation, $\hat{\mathbf{s}}$, with performance measured by a distortion function that explicitly depends on some distortion side information $\mathbf{q}$. For example, one possible distortion function is $d(s, \hat{s}; q) = q \cdot (s - \hat{s})^2$ where the error between the source and its reconstruction are weighted by the distortion side information.

Intuitively, the encoder should tailor its description of the source to the distortion side information. If $\mathbf{q}$ is known at both the encoder and the decoder, this is a simple matter. But in many scenarios, full knowledge of the distortion side information may be unavailable. Hence we explore fundamental limits and practical coding schemes when $\mathbf{q}$ is known only at the encoder, only at the decoder, both, or neither.

We show that such distortion side information is not only useful at the encoder, but that under certain conditions knowing it at the encoder is as good as knowing it everywhere and knowing it at the decoder is useless. Thus distortion side information is a natural complement to the signal side information studied by Wyner and Ziv [189] which depends on the source but does not affect the distortion measure. Furthermore, when both types of side information are present, we show that knowing the distortion side information only at the encoder and knowing the signal side information only at the decoder is asymptotically as good as complete knowledge of all side information. Finally, we characterize the penalty for not providing a given type of side information at the right place and show it can be arbitrarily large.

### ■ 1.1.2 Fixed Lag Signal Side Information

Most current analysis of the value of side information considers non-causal signal side information. But in many real-time applications such as control or sensor networks, signal side information is often available causally with a fixed lag. For example, consider a remote sensor that sends its observations to a controller as illustrated in Fig. 1-1 (reproduced from Chapter 5). The sensor may be a satellite or aircraft reporting the upcoming temperature, wind speed, or other weather data to a vehicle. The sensor observations must be encoded via lossy compression to conserve power or bandwidth. In contrast to the standard lossy compression scenario, however, the controller directly observes the original, uncompressed data after some delay. The goal of the sensor is to provide the controller with information about upcoming events *before* they occur. Thus at first it might not seem that observing the true, uncompressed data *after* they occur would be useful.

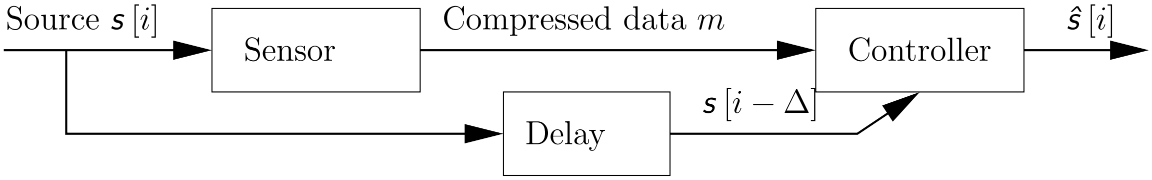


**Figure 1-1.** A sensor compresses and sends the source sequence $s[1]$, $s[2]$, ..., to a controller which reconstructs the quantized sequence $\hat{s}[1]$, $\hat{s}[2]$, ..., in order to take some control action. After a delay or lag of $\Delta$, the controller observes the original, uncompressed data directly.

In Chapter 5 we study how these delayed observations of the source data can be used. Our main result is that such information can be quite valuable. Specifically, we focus on the special case of perfect side information with unit lag corresponding to source coding with feedforward (the dual of channel coding with feedback) introduced by Pradhan [144]. We use this duality to develop a linear complexity algorithm which achieves the rate-distortion bound for any memoryless finite alphabet source and distortion measure.

## ■ 1.2 Delay Constraints

Delay is of fundamental importance in many systems. In control, delay directly affects stability. For interactive applications such as Voice over IP (VoIP), video-conferencing, and distance learning, delays become noticeable and disturbing once they exceed a few hundred milliseconds. Even for non-interactive tasks such as Video-on-Demand and other streaming media applications, latency is a key issue. Some of the existing work on delay and latency includes the design and analysis of protocols aimed at minimizing delay or latency due to congestion in wired networks such as the Internet [116, 15, 61, 113, 8, 26, 88, 22, 39, 182, 90]. Essentially, such research studies how to efficiently allocate resources and control queues to minimize congestion when possible and fairly distribute such congestion when it is unavoidable. Even with the best protocols, algorithms, and architectures, the decentralized nature of communication networks implies that aggressive use of shared resources will result in some amount of dropped or delayed packets just as increased utilization of a queue will result in longer service times.

Other work considers delay in various forms through the study of error correction codes and diversity schemes to provide robustness to time-varying channel fluctuations such as signal fading, shadowing, mobility, congestion, and interference [141, 35, 23, 93, 204, 97, 135, 28, 7, 6, 5, 32]. Traditional methods of providing reliable communication usually rely on powerful error correcting codes, interleaving, or other forms of diversity to achieve robustness. While such traditional methods are highly suited to minimizing the resources required to deliver long, non-causal messages (*e.g.*, file transfers), they are not necessarily suitable for minimizing delay in streaming applications.

In contrast to previous work, this thesis focuses on how to encode information streams to allow fast recovery from dynamic channel fluctuations. Thus the key difference from protocol design is in the use of coding.[1] The key difference between previous coding approaches and this work is the focus on the delay required to continuously decode a message stream as opposed to the delay required to decode one message block. To illustrate the new paradigm, consider the difference in transmitting a file versus broadcasting a lecture in real-time. One portion of a file may be useless without the rest. Hence, in sending a file, delay is naturally measured as the time between when transmission starts and ends. Performance for file transfers can thus be measured as a function of the total transmission time *e.g.*, as studied in the analysis of error exponents.

By contrast, a lecture is intended to be viewed or heard sequentially. Hence, delay is naturally measured as the lag between when the lecturer speaks and the listener hears. The minimum, maximum, or average lag is thus a more appropriate measure than the total transmission time. This thesis addresses these issues by examining fundamental limits and developing practical techniques for data compression, error correction, and network management when delay is a limited, finite resource. A brief discussion of several facets of the general system design problem follow.

### ■ 1.2.1 Streaming Codes For Bursty Channels

In many scenarios, communication suffers from occasional bursts of reduced quality. In the Internet, bursts of packet losses can occur when router queues overflow, while in wireless applications interference from a nearby transmitter or attenuation due to movement (*e.g.*, fading or shadowing) can both hamper reception. How should a system be designed to allow quick recovery from bursts without incurring too much overhead during periods of nominal operation?

One common approach to dealing with bursts is to use interleaving. However, their is a trade-off in choosing long interleavers to spread data symbols out beyond the burst and choosing short interleavers to limit delay. Furthermore certain coding structures may be better matched to a particular interleaver than others [99]. Finally, there is no reason to believe that simply interleaving existing coding schemes is optimal.

[1] Of course, both protocol design and coding are important and ideally should be considered together.

Hence, before evaluating the performance of existing systems, it is useful to characterize the fundamental performance limits in the presence of bursts. Using tools from previous work on bursty channels [62] [69], we derive an upper bound on the communication rate as a function of the burstiness and delay. For example, consider a system which must correct any burst of $\mathtt{B}$ erased packets with a decoding delay of $\mathtt{T}$. We show that such a system must have rate at most

$$R \leq \frac{\mathtt{T}}{\mathtt{T}+\mathtt{B}} \tag{1.1}$$

in the sense that a fraction $1-R$ of each packet must carry redundant information (*e.g.*, parity check bits).

Alternatively, consider a system where occasionally a burst of packets suffers a higher error rate (*e.g.*, due to fading or interference). Specifically, imagine that nominally packets are received without error, but occasionally a burst of $\mathtt{B}$ packets are received where each bit in the packet may be flipped with probability $p$. If any such burst can be corrected with delay $\mathtt{T}$, then our bounds indicate that the system must have rate at most

$$R \leq \frac{\mathtt{T}}{\mathtt{T}+\mathtt{B}} + \frac{\mathtt{B}\cdot[1-H_b(p)]}{\mathtt{T}+\mathtt{B}}$$

where $H_b(p)$ is the binary entropy function.

Generally, a system which must correct any burst of $\mathtt{B}$ degraded packets with a decoding delay of $\mathtt{T}$ must have rate at most

$$R \leq \frac{\mathtt{T}\cdot I(\mathsf{x};\mathsf{y}|\theta_{\mathcal{G}})}{\mathtt{T}+\mathtt{B}} + \frac{\mathtt{B}\cdot I(\mathsf{x};\mathsf{y}|\theta_{\mathcal{B}})}{\mathtt{T}+\mathtt{B}} \tag{1.2}$$

where $I(\mathsf{x};\mathsf{y}|\theta_{\mathcal{G}})$ and $I(\mathsf{x};\mathsf{y}|\theta_{\mathcal{B}})$ specify the mutual information for nominal and degraded packets respectively. Optimal codes which achieve this trade-off can be designed by combining a special kind of incremental coding with a matched interleaver and incremental decoding.

While our general construction for arbitrary channel models is based on a random coding argument (and hence too complex to implement directly), it suggests efficient practical implementations. Specifically, our random coding construction illustrates a direct connection between the structure of good codes and the dynamics of the channel. This connection suggests that instead of designing codes to optimize minimum distance, product distance, or similar static measures of performance, code design must explicitly take into account the channel dynamics to achieve low decoding delays. For example, the code structure in Fig. 1-2 [119] achieves the minimum delay required to correct bursts of a given length and illustrates the connection between channel dynamics and code design. These issues are discussed in more detail in Chapter 7 and practical code constructions are considered in Chapter 8.

## ■ 1.2.2 Delay Universal Streaming Codes

While bursts of packet losses, interference, fading, *etc.*, are common channel impairments they comprise only a small class of all possible channel degradations. In some applications, a system may need to be robust to a variety of channel conditions. For example, in a broadcast or multicast system, different users may receive the same signal over different channel conditions and the best possible decoding lag may be different for each user depending on the received channel quality. Even for a single user, the best possible lag may vary

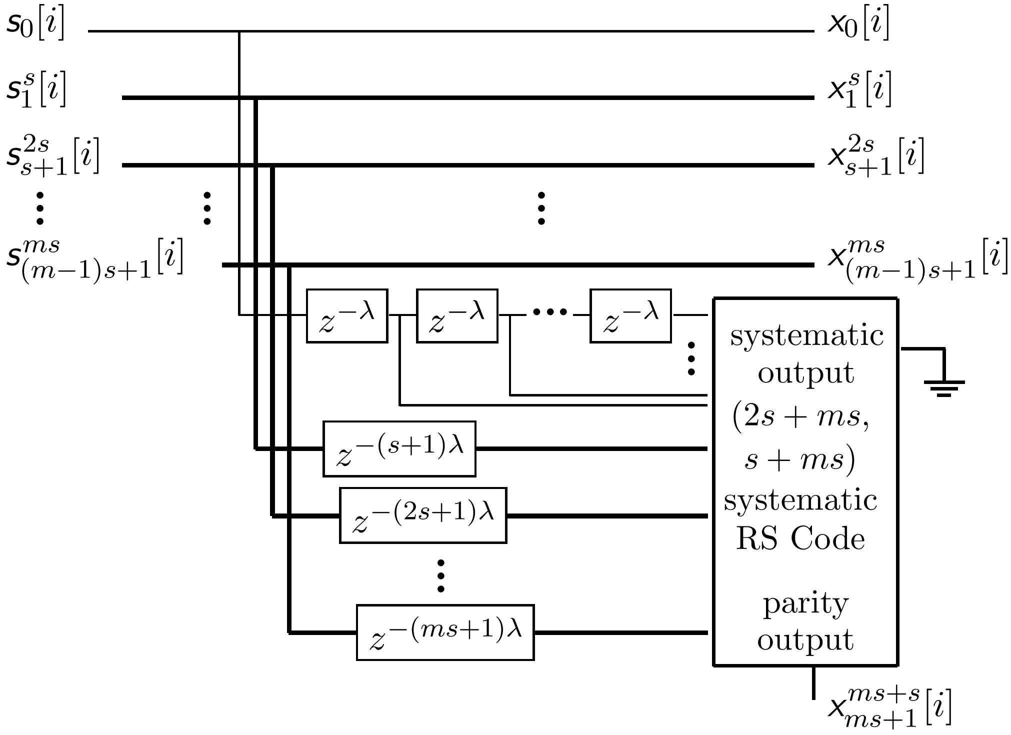


**Figure 1-2.** An encoding structure designed to correct bursts of $\lambda s$ consecutive lost packets with delay $\lambda(ms+1)$ with a rate $(ms+1)/(ms+1+s)$ code where $\lambda$, $m$, and $s$ can take any non-negative integer values chosen by the system designer. The $i$th message, $\mathbf{s}[i]$, is divided into $ms+1$ equal size pieces (denoted $\mathbf{s}_0[i]$, $\mathbf{s}_1[i]$, …, $\mathbf{s}_{ms}[i]$), interleaved, encoded with a systematic Reed-Solomon code and collected into $ms+1+s$ pieces to produce the coded packet $\mathbf{x}[i]$. Bold lines/symbols indicate vectors. Note that the code parameters, $m$, $s$, and $\lambda$, are directly connected to the decoding delay and correctable burst length.

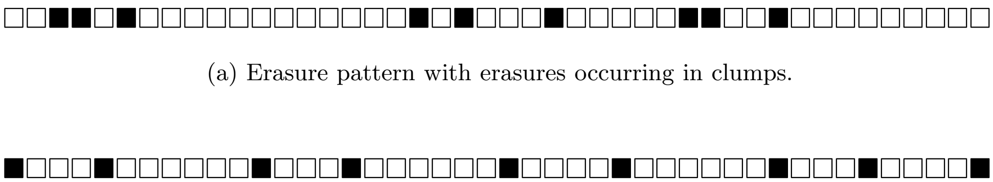
throughout the transmission as the channel varies. Although, traditional codes may be applied to such scenarios, they may be far from optimal. Hence a channel code which is *good* for a large class of degradations (*e.g.*, as opposed to being *optimal* only for bursts) may be desirable.

Our goal is to compute the best possible delays or lags and design systems which achieve them. Essentially, we wish to study transmission of a single message which can be received over a collection of channel ensembles denoted by $\{\mathbf{\Theta}_i\}$ and defined in detail in Section 9.2 of Chapter 9. The decoding delay will of course depend upon the particular channel conditions encountered. But, codes which minimize delay for $\mathbf{\Theta}$ may be poor for $\mathbf{\Theta}'$. Therefore, achieving good overall performance depends on designing codes which perform well for each $\mathbf{\Theta} \in \{\mathbf{\Theta}_i\}$.

For example, in a multicast scenario, user $i$ may receive the signal over channel ensemble $\mathbf{\Theta}_i$. Similarly, even with only a single receiver, there may be some uncertainty about the channel or the channel may act differently at different times and each $\mathbf{\Theta}_i$ may correspond to one such possible channel. Ideally, we would like a system which achieves the minimum possible decoding delay for each possible channel condition.

Robust communication over different types of channels has been studied with the traditional broadcast channel model [43], the compound channel model [101, 187, 25] and more recently the static broadcast model [163,164] and digital fountain codes [31,32,33,114]. Essentially, these models all consider scenarios with different messages, mutual information, or optimal channel input distributions for the different receivers. In contrast, we are interested in channels with different dynamics.

For example, as illustrated by the two erasure channel output sequences in Fig. 1-3 (reproduced from Chapter 9), two channels may be able to support the same information rate, but have different dynamics. Specifically, even though the channel outputs in Fig. 1-3(a) and Fig. 1-3(b) have the same number of erasures, these erasures are clumped together in the former and spread out in the latter. Intuitively, these different kinds of channel dynamics may result in different delays or require different coding structures.



(a) Erasure pattern with erasures occurring in clumps.

(b) Erasure pattern with erasures occurring far apart.

**Figure 1-3.** Channels with same number of erasures but different dynamics.

To illustrate the shortcomings of using traditional block codes, consider a packet loss channel which behaves in one of two possible modes. In the first mode corresponding to Fig. 1-3(b) and denoted $\mathbf{\Theta}_1$, no more than 1 packet is lost within a given window of time. In the second mode corresponding to Fig. 1-3(a) and denoted $\mathbf{\Theta}_2$, up to 3 packets may be lost in the time window of interest.

With a traditional block code, the transmitter could encode each group of 9 source packets, $\mathbf{s}[0]$, $\mathbf{s}[1]$, …, $\mathbf{s}[8]$ into 12 coded packets, $\mathbf{x}[0]$, …, $\mathbf{x}[11]$ using a systematic $(12,9)$ Reed-Solomon (RS) code (or any other equivalent Maximum Distance Separable code). With this approach, any pattern of 3 (or less) packet losses occurring for channel $\mathbf{\Theta}_2$ can be corrected. But, on channel $\mathbf{\Theta}_1$, if only $\mathbf{x}[0]$ is lost, the soonest it can be recovered is when 9 more coded packets are received. Evidently the system may incur a decoding delay of 9 packets just to correct a single packet loss.

To decrease the delay, instead of using a $(12,9)$ code, each block of 3 source packets could be encoded into 4 coded packets using a $(4,3)$ RS code. Since one redundant packet is generated for every three source packets, this approach requires the same redundancy as the $(12,9)$ code. With the $(4,3)$ code, however, if the channel is in mode $\mathbf{\Theta}_1$ and only $\mathbf{x}[0]$ is lost, it can be recovered after the remaining 3 packets in the block are received. Thus the $(4,3)$ system incurs a delay of only 3 to correct one lost packet. While this delay is much smaller than with the $(12,9)$ code, if more than one packet in a block is lost on channel $\mathbf{\Theta}_2$, then decoding is impossible with the $(4,3)$ code.

Both practical block codes as well as traditional information theoretic arguments are not designed for real-time systems. Thus we see that minimizing delay for minor losses from $\mathbf{\Theta}_1$ and maximizing robustness for major losses from $\mathbf{\Theta}_2$ are conflicting objectives. Is this trade-off fundamental to the nature of the problem, or is it an artifact of choosing a poor code structure? We show that in many cases of practical interest, there exist better code structures which are universally optimal for all channel conditions. Specifically, for the packet loss example, there exist codes with both the low decoding delay of the $(4,3)$ code and the robustness of the $(12,9)$ code. Chapter 9 presents a more detailed discussion of these issues.

### ■ 1.2.3 Low Delay Application and Physical Layer Diversity Architectures

Sections 1.2.1 and 1.2.2 focus on coding information over multiple blocks or packets. The goal is to spread information over time so that if some blocks suffer significant channel impairments, the corresponding components of the message stream can be quickly recovered from later blocks. Furthermore, the techniques outlined in Sections 1.2.1 and Section 1.2.2 focus on the properties of the communication channel and are essentially independent of the source.

While Shannon showed that separate source and channel coding are asymptotically optimal when delays are ignored, separation no longer holds in the presence of delay constraints. Hence, as a complement to the previous multi-block coding approaches, we also consider coding approaches for a single-block which simultaneously consider properties of both the source and channel. Essentially, the single-block systems we consider spread the source information over space in order to achieve diversity. Of course, the single-block schemes considering both source coding and channel coding could be combined with the multi-block coding ideas, but we leave this extension for future work.

Diversity techniques for coding over space often arise as appealing means for improving performance of multimedia communication over certain types of channels with independent parallel components (*e.g.*, multiple antennas, frequency bands, time slots, or transmission paths). As illustrated in Fig. 1-4 (a) (reproduced from Chapter 10), diversity can be obtained by channel coding across parallel components at the physical layer, a technique we call *channel coding diversity.* Alternatively as illustrated in Fig. 1-4 (b), the physical layer can present an interface to the parallel components as separate, independent links thus allowing the application layer to implement diversity in the form of multiple description source coding, a technique we call *source coding diversity.* We compare these two approaches in terms of average end-to-end distortion as a function of channel signal-to-noise ratio (SNR).

For on-off channel models, source coding diversity achieves better performance. For channels with a continuous range of reception quality, we show the reverse is true. Specifically, we introduce a new figure of merit called the distortion exponent which measures how fast the average distortion decays with SNR. For continuous models such as additive white Gaussian noise channels with multiplicative Rayleigh fading, our analysis shows that optimal channel coding diversity at the physical layer is more efficient than source coding diversity at the application layer. In particular, we quantify the performance gap between optimal channel coding diversity at the physical layer and other system architectures by computing the distortion exponent in a variety of scenarios.

Finally, we partially address joint source-channel diversity by examining source diversity with joint decoding, an approach in which the channel decoders essentially take into account correlation between the multiple descriptions. We show that by using joint decoding, source coding diversity achieves the same distortion exponent as systems with optimal channel coding diversity. Combined with the advantages of source-coding diversity for on-off channels, this result quantifies the performance advantage of joint source-channel decoding over exploiting diversity either at only the application layer or at only the physical layer. Details are discussed in Chapter 10.

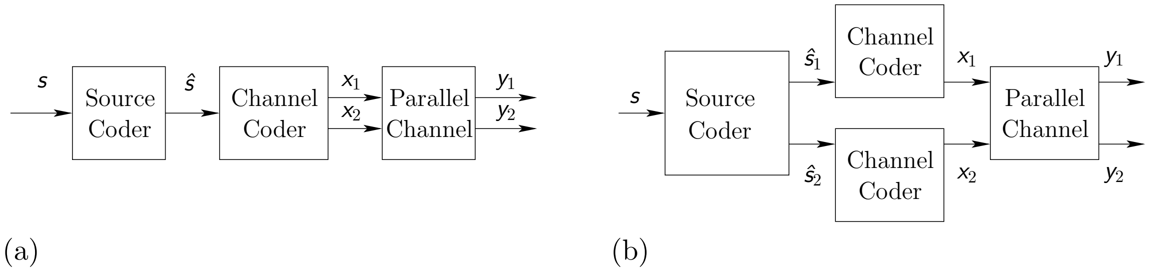


**Figure 1-4.** Diagrams for (a) channel coding diversity and (b) source coding diversity.

## ■ 1.3 Outline of the Thesis

We consider distributed source coding issues in Part I and delay constraints in Part II. The main notation used throughout the thesis is summarized in Appendix A, but most chapters describe the essential notation to allow separate reading of each chapter.

Chapter 2 introduces the idea of distortion side information and derives fundamental performance limits using information-theoretic tools. In Chapter 3 we develop practical quantizers based on lattice codes to efficiently use distortion side information. Developing low complexity source codes which approach fundamental limits is generally difficult. Hence, motivated by the recent success of error correcting codes based on graphical models, we consider iterative quantization using codes on graphs in Chapter 4. The resulting codes can be used as a key component of the quantizers in Chapter 3. Finally, Chapter 5 considers the source coding with fixed lag side information problem discussed in Section 1.1.2.

In Chapter 6 we outline the system model we use for analyzing delay in streaming applications and discuss its relationship to previous work. Next, in Chapter 7, we develop the theory of low delay burst correction outlined in Section 1.2.1. Chapter 8 considers practical constructions of such codes. Chapter 9 explores low delay codes for more general channels as discussed in Section 1.2.2. In Chapter 10, we compare physical-layer and application-layer approaches to obtain diversity to a joint-source channel coding approach as discussed in Section 1.2.3. Finally, we close with some concluding remarks in Chapter 11. Most proofs are provided in the appendices.

The two parts of the thesis are largely independent and can be read separately. Within the parts, Chapter 3 depends somewhat upon Chapter 2 but a survey of the first few sections of Chapter 2 should be sufficient to read Chapter 3. sufficient in reading the latter. Both Chapter 4 and Chapter 5 can be read independently of other chapters. Chapter 7, Chapter 8 and Chapter 9 are somewhat related and a survey of the first part of Chapter 7 will be useful in reading the other two. Chapter 10 can be read independently.

# Part I

# Distributed Information In Source Coding

Chapter 2

# Distortion Side Information

## ■ 2.1 Introduction

In many large systems such as sensor networks, communication networks, and biological systems different parts of the system may each have limited or imperfect information but must somehow cooperate. Key issues in such scenarios include the penalty incurred due to the lack of shared information, possible approaches for combining information from different sources, and the more general question of how different kinds of information can be partitioned based on the role of each system component.

One example of this scenario is when an observer records a signal $\mathbf{s}$ to be conveyed to a receiver who also has some additional signal side information $\mathbf{w}$ which is correlated with $\mathbf{s}$. As demonstrated by various researchers, in many cases the observer and receiver can obtain the full benefit of the signal side information even if it is known only by the receiver [44, 50, 190].

In this chapter we consider a different scenario where instead the observer has some distortion side information $\mathbf{q}$ which describes what components of the data are more sensitive to distortion than others, but the receiver may not have access to $\mathbf{q}$. Specifically, let us model the differing importance of different signal components by measuring the distortion between the $i$th source sample, $s[i]$, and its quantized value, $\hat{s}[i]$, by a distortion function which depends on the side information $q[i]$: $d(s[i], \hat{s}[i]; q[i])$.

In principle, one could treat the source-side information pair $(\mathbf{q}, \mathbf{s})$ as an "effective composite source", and apply conventional techniques to quantize it. Such an approach, however, ignores the different effect $\mathbf{q}$ and $\mathbf{s}$ have on the distortion. And as often happens in lossy compression, a good understanding of the distortion measure may lead to better designs.

Sensor observations are on class of signals where the idea of distortion side information may be useful. For example, a sensor may have side information corresponding to reliability estimates for measured data (which may or may not be available at the receiver). This may occur if the sensor can calibrate its accuracy to changing conditions (*e.g.*, the amount of light, background noise, or other interference present), if the sensor averages data for a variety of measurements (*e.g.*, combining results from a number of sub-sensors) or if some external signal indicates important events (*e.g.*, an accelerometer indicating movement).

Alternatively, certain components of the signal may be more or less sensitive to distortion due to masking effects or context [85]. For example errors in audio samples following a loud sound, or errors in pixels spatially or temporally near bright spots are perceptually less relevant. Similarly, accurately preserving certain edges or textures in an image or human voices in audio may be more important than preserving background patterns/sounds. Masking, sensitivity to context, etc., is usually a complicated function of the entire signal. Yet often there is no need to explicitly convey information about this function to the encoder. Hence, from the point of view of quantizing a given sample, it is reasonable to model such effects as side information.

Clearly in performing data compression with distortion side information, the encoder should weight matching the more important data more than matching the less important data. The importance of exploiting the different sensitivities of the human perceptual system are widely recognized by engineers involved in the construction and evaluation of practical compression algorithms *when distortion side information is available at both observer and receiver*. In contrast, the value and use of distortion side information known only at either the encoder or decoder but not both has received relatively little attention in the information theory and quantizer design community. The rate-distortion function with decoder-only side information, relative to side information dependent distortion measures (as an extension of the Wyner-Ziv setting [190]), is given in [50]. A high resolution approximation for this rate-distortion function for locally quadratic weighted distortion measures is given in [110].

We are not aware of an information-theoretic treatment of encoder-only side information with such distortion measures. In fact, the mistaken notion that encoder only side information is never useful is common folklore. This may be due to a misunderstanding of Berger's result that side information *which does not affect the distortion measure* is never useful when known only at the encoder [19].

In this chapter, we begin by studying the rate-distortion trade-off when side information about the distortion sensitivity is available. We show that such distortion side information can provide an arbitrarily large advantage (relative to no side information) even when the distortion side information is known only at the encoder. Furthermore, we show that just as knowledge of signal side information is often only required at the decoder, knowledge of distortion side information is often only required at the encoder. Finally, we show that these results continue to hold even when both distortion side information $\mathbf{q}$ and signal side information $\mathbf{w}$ are considered. Specifically, we demonstrate that a system where only the encoder knows $\mathbf{q}$ and only the decoder knows $\mathbf{w}$ is asymptotically as good as a system with all side information known everywhere. We also derive the penalty for deviating from this side information configuration (*e.g.*, providing $\mathbf{q}$ to the decoder instead of the encoder).

We first illustrate how distortion side information can be used even when known only by the observer with some examples in Section 2.2. Next, in Section 2.3, we precisely define a problem model and state the relevant rate-distortion trade-offs. In Section 2.4 we discuss specific scenarios where encoder only distortion side information is just as good as full distortion side information. In Section 2.5, we study more general source and distortion models in the limit of high-resolution. Specifically, we show that in high-resolution, knowing distortion side information at the encoder and signal side information at the decoder is both necessary and sufficient to achieve the performance of a fully informed system. To illustrate how quickly the high-resolution regime is approached we focus on the special case of scaled quadratic distortions in Section 2.6. Finally, we close with a discussion in Section 2.7 followed by some concluding remarks in Section 2.8. Throughout the chapter, most proofs and lengthy derivations are deferred to the appendix.

## ■ 2.2 Examples

### ■ 2.2.1 Discrete Uniform Source

Consider the case where the source, $\mathsf{s}[i]$, corresponds to $n$ samples each uniformly and independently drawn from the finite alphabet $\mathcal{S}$ with cardinality $|\mathcal{S}| \geq n$. Let $\mathsf{q}[i]$ correspond to $n$ binary variables indicating which source samples are relevant. Specifically, let the distortion measure be of the form $d(s, \hat{s}; q) = 0$ if and only if either $q = 0$ or $s = \hat{s}$. Finally, let the sequence $\mathsf{q}[i]$ be statistically independent of the source with $\mathsf{q}[i]$ drawn uniformly from the $n$ choose $k$ subsets with exactly $k$ ones.

If the side information were unavailable or ignored, then losslessly communicating the source would require exactly $n \cdot \log |\mathcal{S}|$ bits. A better (though still sub-optimal) approach when encoder side information is available would be for the encoder to first tell the decoder which samples are relevant and then send only those samples. This would require $n \cdot H_b(k/n) + k \cdot \log |\mathcal{S}|$ bits where $H_b(\cdot)$ denotes the binary entropy function. Note that if the side information were also known at the decoder, then the overhead required in telling the decoder which samples are relevant could be avoided and the total rate required would only be $k \cdot \log |\mathcal{S}|$. We will show that this overhead can in fact be avoided even without decoder side information.

Pretend that the source samples $\mathsf{s}[0]$, $\mathsf{s}[1]$, $\ldots$, $\mathsf{s}[n-1]$, are a codeword of an $(n, k)$ Reed-Solomon (RS) code (or more generally any MDS[1] code) with $\mathsf{q}[i] = 0$ indicating an erasure at sample $i$. Use the RS *decoding* algorithm to "correct" the erasures and determine the $k$ corresponding information symbols which are sent to the receiver. To reconstruct the signal, the receiver *encodes* the $k$ information symbols using the encoder for the $(n, k)$ RS code to produce the reconstruction $\hat{\mathsf{s}}[0]$, $\hat{\mathsf{s}}[1]$, $\ldots$, $\hat{\mathsf{s}}[n-1]$. Only symbols with $\mathsf{q}[i] = 0$ could have changed, hence $\hat{\mathsf{s}}[i] = \mathsf{s}[i]$ whenever $\mathsf{q}[i] = 1$ and the relevant samples are losslessly communicated using only $k \cdot \log |\mathcal{S}|$ bits.

As illustrated in Fig. 2-1, RS decoding can be viewed as curve-fitting and RS encoding can be viewed as interpolation. Hence this source coding approach can be viewed as fitting a curve of degree $k-1$ to the points of $\mathsf{s}[i]$ where $\mathsf{q}[i] = 1$. The resulting curve can be specified using just $k$ elements. It perfectly reproduces $\mathsf{s}[i]$ where $\mathsf{q}[i] = 1$ and interpolates the remaining points.

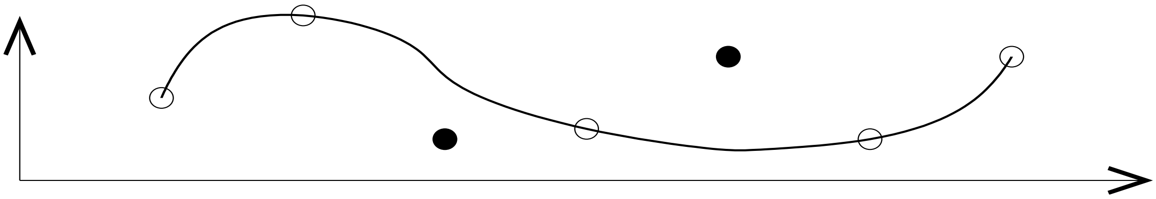


**Figure 2-1.** Losslessly encoding a source with $n = 7$ points where only $k = 5$ points are relevant (*i.e.*, the unshaded ones), can be done by fitting a fourth degree curve to the relevant points. The resulting curve will require $k$ elements (yielding a compression ratio of $k/n$) and will exactly reproduce the desired points.

### 2.2.2 Gaussian Source

A similar approach can be used to quantize a zero mean, unit variance, complex Gaussian source relative to quadratic distortion using the Discrete Fourier Transform (DFT). Specifically, to encode the source samples $\mathsf{s}[0]$, $\mathsf{s}[1]$, $\ldots$, $\mathsf{s}[n-1]$, pretend that they are samples of a complex, periodic, Gaussian, sequence with period $n$, which is band-limited in the sense that only its first $k$ DFT coefficients are non-zero. Using periodic, band-limited, interpolation we can use only the $k$ samples for which $\mathsf{q}[i] = 1$ to find the corresponding $k$ DFT coefficients, $\mathsf{S}[0]$, $\mathsf{S}[1]$, $\ldots$, $\mathsf{S}[k-1]$.

The relationship between the $k$ relevant source samples and the $k$ interpolated DFT coefficients has a number of special properties. In particular this $k \times k$ transformation is unitary. Hence, the DFT coefficients are Gaussian with unit variance and zero mean. Thus, the $k$ DFT coefficients can be quantized with average distortion $D$ per coefficient and $k \cdot R(D)$ bits where $R(D)$ represents the rate-distortion trade-off for the quantizer. To reconstruct the signal, the decoder simply transforms the quantized DFT coefficients back to the time domain. Since the DFT coefficients and the relevant source samples are related

[1] The desired MDS code always exists since we assumed $|\mathcal{S}| \geq n$. For $|\mathcal{S}| < n$, near MDS codes exist which give asymptotically similar performance with an overhead that goes to zero as $n \to \infty$.

by a unitary transformation, the average error per coefficient for these source samples is exactly $D$.

Note if the side information were unavailable or ignored, then at least $n \cdot R(D)$ bits would be required. If the side information were losslessly sent to the decoder, then $n \cdot H_b(k/n) + k \cdot R(D)$ would be required. Finally, even if the decoder had knowledge of the side information, at least $k \cdot R(D)$ bits would be needed. Hence, the DFT scheme achieves the same performance as when the side information is available at both the encoder and decoder, and is strictly better than ignoring the side information or losslessly communicating it.

## ■ 2.3 Problem Model

Vectors and sequences are denoted in bold (*e.g.*, $\mathbf{x}$) with the $i$th element denoted as $x[i]$. Random variables are denoted using the sans serif font (*e.g.*, $\mathsf{x}$) while random vectors and sequences are denoted with bold sans serif (*e.g.*, $\boldsymbol{\mathsf{x}}$). We denote mutual information, entropy, and expectation as $I(\mathsf{x};\mathsf{y})$, $H(\mathsf{x})$, $E[\mathsf{x}]$. Calligraphic letters denote sets (*e.g.*, $s \in \mathcal{S}$).

We are primarily interested in two kinds of side information which we call "signal side information" and "distortion side information". The former (denoted $\boldsymbol{\mathsf{w}}$) corresponds to information which is statistically related to the source but does not directly affect the distortion measure and the latter (denoted $\boldsymbol{\mathsf{q}}$) corresponds to information which is not directly related to the source but does directly affect the distortion measure. This decomposition proves useful since it allows us to isolate two important insights in source coding with side information. First, as Wyner and Ziv discovered [190], knowing $\boldsymbol{\mathsf{w}}$ only at the decoder is often sufficient. Second, as our examples in Section 2.2 illustrate, knowing $\boldsymbol{\mathsf{q}}$ only at the encoder is often sufficient. Furthermore, the relationship between the side information and the distortion measure and the relationship between the side information and the source often arise from physically different effects and so such a decomposition is warranted from a practical standpoint. Of course, such a decomposition is not always possible and we explore some issues for general side information, $\boldsymbol{\mathsf{z}}$, which affects both the source and distortion measure in Sections 2.5.3 and 2.7.3.

In any case, we define the source coding with side information problem as the tuple

$$(\mathcal{S}, \hat{\mathcal{S}}, \mathcal{Q}, \mathcal{W}, p_{\mathsf{s}}(s), p_{\mathsf{w}|\mathsf{s}}(w|s), p_{\mathsf{q}|\mathsf{w}}(q|w), d(\cdot,\cdot;\cdot)). \tag{2.1}$$

Specifically, a source sequence $\boldsymbol{\mathsf{s}}$ consists of the $n$ samples $\mathsf{s}[1]$, $\mathsf{s}[2]$, ..., $\mathsf{s}[n]$ drawn from the alphabet $\mathcal{S}$. The signal side information $\boldsymbol{\mathsf{w}}$ and the distortion side information $\boldsymbol{\mathsf{q}}$ likewise consist of $n$ samples drawn from the alphabets $\mathcal{W}$ and $\mathcal{Q}$ respectively. These random variables are generated according to the distribution

$$p_{\boldsymbol{\mathsf{s}},\boldsymbol{\mathsf{q}},\boldsymbol{\mathsf{w}}}(\mathbf{s}, \mathbf{q}, \mathbf{w}) = \prod_{i=1}^{n} p_{\mathsf{s}}(s[i]) \cdot p_{\mathsf{w}|\mathsf{s}}(w[i] | s[i]) \cdot p_{\mathsf{q}|\mathsf{w}}(q[i] | w[i]). \tag{2.2}$$

Fig. 2-2 illustrates the sixteen possible scenarios where $\boldsymbol{\mathsf{q}}$ and $\boldsymbol{\mathsf{w}}$ may each be available at the encoder, decoder, both, or neither depending on whether the four switches are open or closed. A rate $R$ encoder, $f(\cdot)$, maps a source as well as possible side information to an index $i \in \{1, 2, \ldots, 2^{nR}\}$. The corresponding decoder, $g(\cdot)$, maps the resulting index as well as possible decoder side information to a reconstruction of the source. Distortion for a source $\boldsymbol{\mathsf{s}}$ which is quantized and reconstructed to the sequence $\hat{\boldsymbol{\mathsf{s}}}$ taking values in the

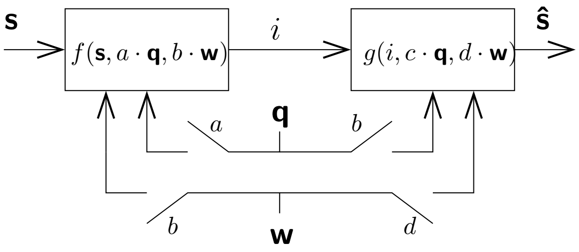

s
$f(\mathbf{s}, a\cdot\mathbf{q}, b\cdot\mathbf{w})$
$i$
$g(i, c\cdot\mathbf{q}, d\cdot\mathbf{w})$
$\hat{\mathbf{s}}$
a
q
b
b
w
d

**Figure 2-2.** Possible scenarios for source coding with distortion dependent and signal dependent side information $\mathbf{q}$ and $\mathbf{w}$. The labels $a$, $b$, $c$, and $d$ are 0/1 if the corresponding switch is open/closed and the side information is unavailable/available to the encoder $f(\cdot)$ or decoder $g(\cdot)$.

alphabet $\hat{\mathcal{S}}$ is measured via

$$d(\mathbf{s}, \hat{\mathbf{s}}; \mathbf{q}) = \frac{1}{n}\sum_{i=1}^{n} d(s[i], \hat{s}[i]; q[i]). \tag{2.3}$$

As usual, the rate-distortion function is the minimum rate such that there exists a system where the distortion is less than $D$ with probability approaching 1 as $n \to \infty$. We denote the sixteen possible rate-distortion functions by describing where the side-information is available. For example, $R[\text{Q-NONE-W-NONE}](D)$ denotes the rate-distortion function without side information and $R[\text{Q-NONE-W-DEC}](D)$ denotes the Wyner-Ziv rate-distortion function where $\mathbf{w}$ is available at the decoder [190]. Similarly, when all information is available at both encoder and decoder $R[\text{Q-BOTH-W-BOTH}](D)$ describes Csiszár and Körner's [50] generalization of Gray's conditional rate-distortion function $R[\text{Q-NONE-W-BOTH}](D)$ [77] to the case where the side information can affect the distortion measure. As suggested to us by Berger [18], all the rate-distortion functions may be derived by considering $\mathbf{q}$ as part of $\mathbf{s}$ or $\mathbf{w}$ (*i.e.*, by considering the "super-source" $\mathbf{s}' = (\mathbf{s}, \mathbf{q})$ or the "super-side-information" $\mathbf{w}' = (\mathbf{w}, \mathbf{q})$) and applying well-known results for source coding, source coding with side information, the conditional rate-distortion theorem, *etc.*

## 2.4 Finite Resolution Quantization

To obtain more precise results regarding the use of distortion side information at the encoder at finite rates and distortions, we generalize the examples discussed in the introduction by focusing on the case where the signal side information, $\mathbf{w}$, is null. With this simplification, the relevant rate-distortion functions are easy to obtain.

**Proposition 1.** *The rate-distortion functions when* $\mathbf{w}$ *is null are*

$$R[\text{Q-NONE-W-NULL}](D) = \inf_{p_{\hat{s}|s}(\hat{s}|s): E[d(s,\hat{s};q)] \le D} I(s; \hat{s}) \tag{2.4a}$$

$$R[\text{Q-DEC-W-NULL}](D) = \inf_{p_{u|s}(u|s), v(\cdot,\cdot): E[d(s,v(u,q);q)] \le D} I(s; u) - I(u; q) \tag{2.4b}$$

$$R[\text{Q-ENC-W-NULL}](D) = \inf_{p_{\hat{s}|s,q}(\hat{s}|s,q): E[d(s,\hat{s};q)] \le D} I(s, q; \hat{s}) = I(s; \hat{s}|q) + I(\hat{s}; q) \tag{2.4c}$$

$$R[\text{Q-BOTH-W-NULL}](D) = \inf_{p_{\hat{s}|s,q}(\hat{s}|s,q): E[d(s,\hat{s};q)] \le D} I(s; \hat{s}|q). \tag{2.4d}$$

The rate-distortion functions in (2.4a), (2.4b), and (2.4d) follow from standard results (*e.g.*, [19,44,50,77,190]). To obtain (2.4c) we can apply the classical rate-distortion theorem to the "super source" $\mathbf{s}' = (\mathbf{s}, \mathbf{q})$ as suggested by Berger [18].

Comparing these rate-distortion functions immediately yields the following result regarding the rate-loss for lack of distortion side information at the decoder.

**Proposition 2.** *Knowing* $\mathbf{q}$ *only at the encoder is as good as knowing it at both encoder and decoder if and only if* $I(\hat{\mathsf{s}}; \mathsf{q}) = 0$ *for the optimizing* $\mathsf{q}$ *in* (2.4c) *and* (2.4d).

The intuition for this result is illustrated in Fig. 2-3. Specifically, $p_{\hat{\mathsf{s}}|\mathsf{q}}(\hat{s}|q)$ represents the distribution of the codebook. Thus if a different codebook is required for different values of $\mathsf{q}$, then the penalty for knowing $\mathsf{q}$ only at the encoder is exactly the information which the encoder must send to the decoder to allow the decoder to determine the proper codebook. The only way that knowing $\mathsf{q}$ at the encoder can be just as good as knowing it at both is if there exists a fixed codebook which is universally optimal regardless of $\mathsf{q}$.

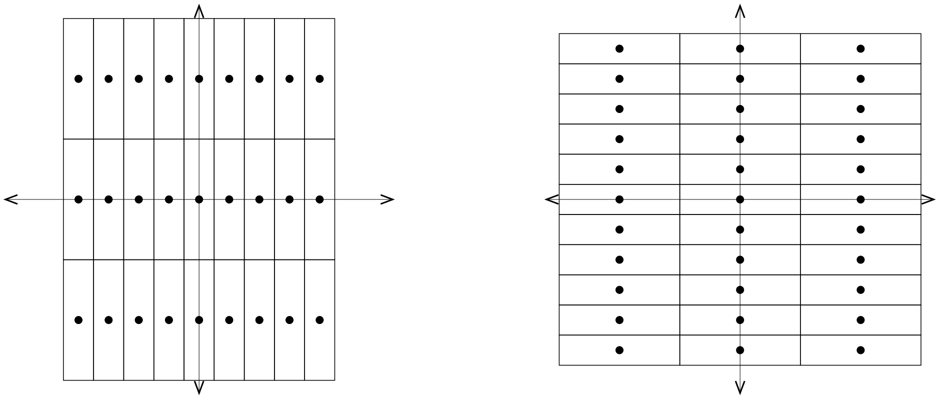


**Figure 2-3.** An example of two different quantizer codebooks for different values of the distortion side information, $\mathsf{q}$. If $\mathsf{q}$ indicates the horizontal error (respectively, vertical error) is more important, then the encoder can use the codebook on the left (resp., right) to increase horizontal accuracy (resp., vertical accuracy). The penalty for knowing $\mathsf{q}$ only at the encoder is the amount of bits required to communicate which codebook was used.

One of the main insights of this chapter is that if a variable partition is allowed, such universally good fixed codebooks exist as illustrated in Fig. 2-4. Specifically $p_{\hat{\mathsf{s}}|\mathsf{q}}(\hat{s}|q)$ represents the distribution of the codebook while $p_{\hat{\mathsf{s}}|\mathsf{s},\mathsf{q}}(\hat{s}|s, q)$ represents the quantizer partition mapping source and side information to a codeword. Thus even if the side information affects the distortion and $p_{\hat{\mathsf{s}}|\mathsf{s},\mathsf{q}}(\hat{s}|s, q)$ depends on $q$, it may be that $p_{\hat{\mathsf{s}}|\mathsf{q}}(\hat{s}|q)$ is independent of $q$. In such cases (characterized by the condition $I(\hat{\mathsf{s}}; \mathsf{q}) = 0$), there exists a *fixed* codebook with a *variable* partition which is simultaneously optimal for all values of the distortion side information $\mathbf{q}$. Specifically, in such a system the reconstruction $\hat{\mathbf{s}}(i)$ corresponding to a particular index $i$ is fixed regardless of $\mathbf{q}$, but the partition mapping $\mathbf{s}$ to the codebook index $i$ depends on $\mathbf{q}$. In various scenarios, this type of fixed codebook variable partition approach can be implemented via a lattice [195].

The quantization error will depend on the source distribution and size of the quantization cells. Thus if the quantization cells of a fixed codebook variable partition system like Fig. 2-4 are the same as the corresponding variable codebook system in Fig. 2-3, the performance will be the same. Intuitively, the main difference between these two figures (as well as

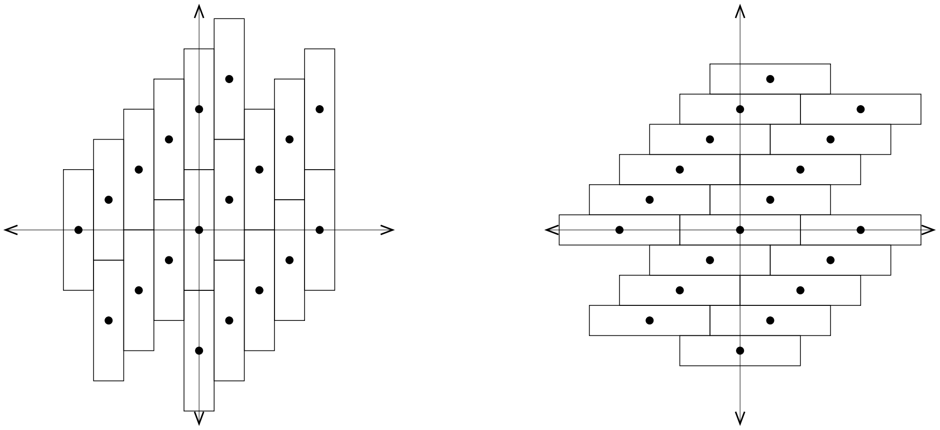


**Figure 2-4.** An example of a quantizer with a variable partition and fixed codebook. If the encoder knows the horizontal error (respectively, vertical error) is more important, it can use the partition on the left (resp., right) to increase horizontal accuracy (resp., vertical accuracy). The decoder only needs to know the quantization point not the partition.

general fixed codebook/variable partition schemes versus variable codebook schemes) is in the jagged tiling of Fig. 2-4 versus the regular tiling of Fig. 2-3. In Sections 2.4.1 and Section 2.4.3, we consider various scenarios where the uniformity in the source or distortion measure makes these two tilings equivalent and thus the condition $I(\hat{\mathsf{s}};\mathsf{q}) = 0$ is satisfied for all distortions. Similarly, in Section 2.5 we show that if we ignore "edge effects" and focus on the high-resolution regime, then the difference in these tilings becomes negligible. Thus in the high-resolution regime we show that a wide range of source and distortion models admit variable partition, fixed codebook quantizers as in Fig. 2-4 and achieve $I(\hat{\mathsf{s}};\mathsf{q}) = 0$.

Finally, we note that Proposition 2 suggests that distortion side information, $\mathsf{q}$, complements signal side information, $\mathsf{w}$, in the sense that $\mathsf{q}$ is useful at the encoder while $\mathsf{w}$ is useful at the decoder. In fact in Section 2.5.2 we strengthen this complementary relationship by showing that $\mathsf{w}$ is often useless at the encoder while $\mathsf{q}$ is often useless at the decoder.

### ■ 2.4.1 Uniform Sources with Group Difference Distortions

Let the source, $\mathsf{s}$, be uniformly distributed over a group, $\mathcal{S}$, with the binary relation, $\oplus$. For convenience, we use the symbol $a \ominus b$ to denote $a \oplus b^{-1}$ (where $b^{-1}$ denotes the additive inverse of $b$ in the group). We define a group difference distortion measure as any distortion measure where

$$d(s,\hat{s};q) = \rho(\hat{s} \ominus s; q) \tag{2.5}$$

for some function $\rho(\cdot;\cdot)$. As we will show, the symmetry in this scenario insures that the optimal codebook distribution is uniform. This allows an encoder to design a fixed codebook and vary the quantization partition based on $\mathsf{q}$ to achieve the same performance as a system where both encoder and decoder know $\mathsf{q}$.

**Theorem 1.** *Let the source be uniformly distributed over a group with a difference distortion measure and let the distortion side information be independent of the source. Then it suffices to choose the test-channel distribution, $p_{\hat{\mathsf{s}}|\mathsf{s},\mathsf{q}}(\hat{s}|s,q)$ such that $p_{\hat{\mathsf{s}}|\mathsf{q}}(\hat{s}|q)$ is uniform (and hence independent of $q$) implying that there is no rate-loss as stated in Proposition 2.*

For either finite or continuous groups we provide the following simple proof using either entropies or differential entropies to give some intuition for why the result holds. For more general "mixed groups" with both discrete and continuous components, entropy is not well defined and a more complicated argument based on symmetry and convexity is provided in Appendix B.1.

*Proof of Theorem 1:* For a finite group, choosing $z^*$ to maximize $H(z|q)$ subject to the constraint $E[\rho(z;q)] \leq D$ yields the following lower bound on $R[\text{Q-ENC-W-NULL}](D)$:

$$\begin{aligned} I(\hat{s};s,q) &= H(s) + H(q) - H(s,q|\hat{s}) && (2.6)\\ &= \log|\mathcal{S}| + H(q) - H(q|\hat{s}) - H(\hat{s}-s|s,q) && (2.7)\\ &\geq \log|\mathcal{S}| - H(\hat{s}-s|q) && (2.8)\\ &\geq \log|\mathcal{S}| - H(z^*|q) && (2.9) \end{aligned}$$

where (2.8) follows since conditioning reduces entropy. Choosing the test-channel distribution $\hat{s} = z^* + s$ achieves this bound with equality and must therefore be optimal. Furthermore, since $\hat{s}$ and $q$ are statistically independent for this test-channel distribution, $I(\hat{s};q) = 0$ and thus comparing (2.4c) and to (2.4d) shows $R[\text{Q-ENC-W-NULL}](D) = R[\text{Q-BOTH-W-NULL}](D)$ for finite groups. The same argument holds for continuous groups with entropy replaced by differential entropy and $|\mathcal{S}|$ replaced by the Lebesgue measure of $\mathcal{S}$. □

## ■ 2.4.2 Examples

Next we outline some cases involving a uniform source and a group difference distortion measure.

### Phase Quantization

One important class of problems where the source is uniformly distributed over a group and distortion is measured via a group difference distortion measure is the phase quantization problem. In applications such as Magnetic Resonance Imaging, Synthetic Aperture Radar, Ultrasonic Microscopy, etc., physical phenomena are inferred from the phase shifts they induce in a probe signal [107,67,52]. Alternatively, when magnitude and phase information must both be recorded, there are sometimes advantages to treating these separately, [138, 84,171]. Finally, it may sometimes be desirable to only record phase information about a signal when unique reconstruction from the phase is possible and the signal magnitude is noisy or otherwise problematic [83].

### Hamming Distortion

Another example of a uniform source with a group difference distortion measure is any problem with a uniform source and Hamming distance distortion. For the case of a binary source we first derive the various rate-distortion trade-offs for a general Hamming distortion measure depending on $q$. Next we adapt this general result to the special cases of quantizing noisy observations and quantizing with a weighted distortion measure. The former demonstrates that the naive encoding method where the encoder losslessly communicates the side information to the decoder and then uses optimal encoding, can require arbitrarily higher *rate* than the optimal rate-distortion trade-off. The second example demonstrates that ignoring the side information can result in arbitrarily higher *distortion* than the optimal encoding method.

**General Formula For Hamming Distortion Depending on q** Consider a symmetric binary source, $\mathsf{s}$, with side information, $\mathsf{q}$, taking values in $\{1, 2, \ldots, N\}$ with distribution $p_{\mathsf{q}}(q)$. Let distortion be measured according to

$$d(s, \hat{s}; q) = \alpha_q + \beta_q \cdot d_H(s, \hat{s}) \tag{2.10}$$

where $\{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ and $\{\beta_1, \beta_2, \ldots, \beta_N\}$ are sets of non-negative weights. In Appendix B.2 we derive the various rate-distortion functions for $D \geq E[\alpha_{\mathsf{q}}]$ to obtain

$$R[\text{Q-NONE-W-NULL}](D) = R[\text{Q-DEC-W-NULL}](D) = 1 - H_b\left(\frac{D - E[\alpha_{\mathsf{q}}]}{E[\beta_{\mathsf{q}}]}\right) \tag{2.11a}$$

$$R[\text{Q-ENC-W-NULL}](D) = R[\text{Q-BOTH-W-NULL}](D) = 1 - \sum_{i=1}^{N} p_{\mathsf{q}}(i) \cdot H_b\left(\frac{2^{-\lambda\beta_i}}{1 + 2^{-\lambda\beta_i}}\right) \tag{2.11b}$$

where $\lambda$ is chosen to satisfy the distortion constraint

$$\sum_{i=1}^{N} p_{\mathsf{q}}(i) \left[\alpha_i + \beta_i \cdot \frac{2^{-\lambda\beta_i}}{1 + 2^{-\lambda\beta_i}}\right] = D. \tag{2.12}$$

**Noisy Observations** To provide a more concrete illustration of the effect of side information at the encoder, consider the special case where $\mathsf{s}$ is a noisy observation of an underlying source received through a binary symmetric channel (BSC) with cross over probability specified by the side information. Specifically, let the cross over probability of the BSC be

$$\epsilon_{\mathsf{q}} = \frac{\mathsf{q} - 1}{2(N - 1)}$$

which is always at most 1/2.

A distortion of 1 is incurred if a cross over occurs due to either the noise in the observation or the noise in the quantization (but not both):

$$\begin{aligned} d(s, \hat{s}; q) &= \epsilon_{\mathsf{q}} \cdot [1 - d_H(s, \hat{s})] + (1 - \epsilon_{\mathsf{q}}) \cdot d_H(s, \hat{s}) \\ &= \epsilon_{\mathsf{q}} + (1 - 2\epsilon_{\mathsf{q}}) \cdot d_H(s, \hat{s}) \\ &= \frac{q - 1}{2(N - 1)} + \left(1 - \frac{q - 1}{N - 1}\right) \cdot d_H(s, \hat{s}). \end{aligned} \tag{2.13}$$

This corresponds to a distortion measure in the form of (2.10) with $\alpha_q = (q - 1)/[2(N - 1)]$ and $\beta_q = 1 - (q - 1)/(N - 1)$. Hence, the rate-distortion formulas from (2.11) apply. The optimal encoding strategy is to encode the noisy observation directly as discussed in [186] although with different amounts of quantization depending on the side information.

In the left plot of Fig. 2-5, we illustrate the rate-distortion function for this problem with and without side information at the encoder in the case where $\mathsf{q} \in \{1, 2\}$ (*i.e.*, $N = 2$) and the observation is either noise-less or completely noisy. In the right plot of Fig. 2-5, we illustrate the rate-distortion functions in the limit where $N \to \infty$ and the noise is uniformly distributed between 0 and 1/2. Note that in the latter case, if the side information were encoded losslessly then $\log N$ extra bits of overhead would be required beyond the amount when optimal encoding is used. Hence, communicating the side information losslessly can require an arbitrarily large rate even though optimal use of the encoder side information reduces the quantization rate.

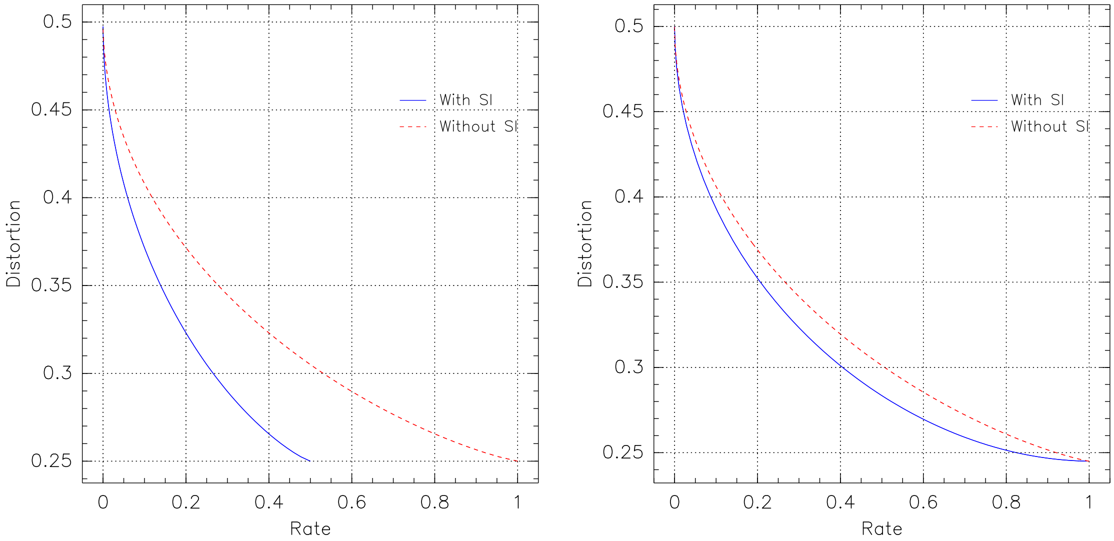


**Figure 2-5.** Rate-Distortion curves for noisy observations of a binary source. The solid curve represents the minimum possible Hamming distortion when side information specifying the cross-over probability of the observation noise is available at the encoder (or both at the encoder and decoder). The dashed curve represents the minimum distortion when side information is not available (or ignored) at the encoder. For the plot on the left the cross over probability for the observation noise is equally likely to be 0 or $1/2$, while for the plot on the right it is uniformly distributed over the interval $[0, 1/2]$.

**Weighted Distortion** In the previous example, certain source samples were more important because they were observed with less noise. In this section we consider a model where certain samples of a source are inherently more important than others (*e.g.*, edges in a binary image, other perceptually important features, or sensor readings in high activity areas). Specifically, we consider a distortion measure of the form (2.10) where $\beta_k = \exp(\gamma k/N)$, $\alpha_k = 0$, and the side information is uniformly distributed over $\{0, 1, \ldots, N-1\}$.

The left plot in Fig. 2-6 illustrates the rate-distortion curves for the case when $N = 2$, while the right plot corresponds to the case where $N \to \infty$. The former model corresponds to two types of samples: important samples where a distortion of $\exp(\gamma/2)$ is incurred if quantized incorrectly and normal samples where a distortion of 1 is incurred if quantized incorrectly. If no side information is available (or if side information is ignored), the encoder must treat these samples equally. If side information at the encoder is used optimally, then more bits are spent quantizing the important samples. The latter model illustrates the case when there is a continuum of importance for the samples.

In the limit as $\gamma \to \infty$ and $N = 2$, the system not using side information, suffers increasingly more distortion. This is most evident for rates greater than $1/2$. In this rate region, the system with side information losslessly encodes the important samples and the distortion is bounded by 0.5 while the system without side information has a distortion which scales with $\exp(\gamma/2)$. Thus the extra distortion required when $\boldsymbol{q}$ is not available to the encoder can be arbitrarily large.

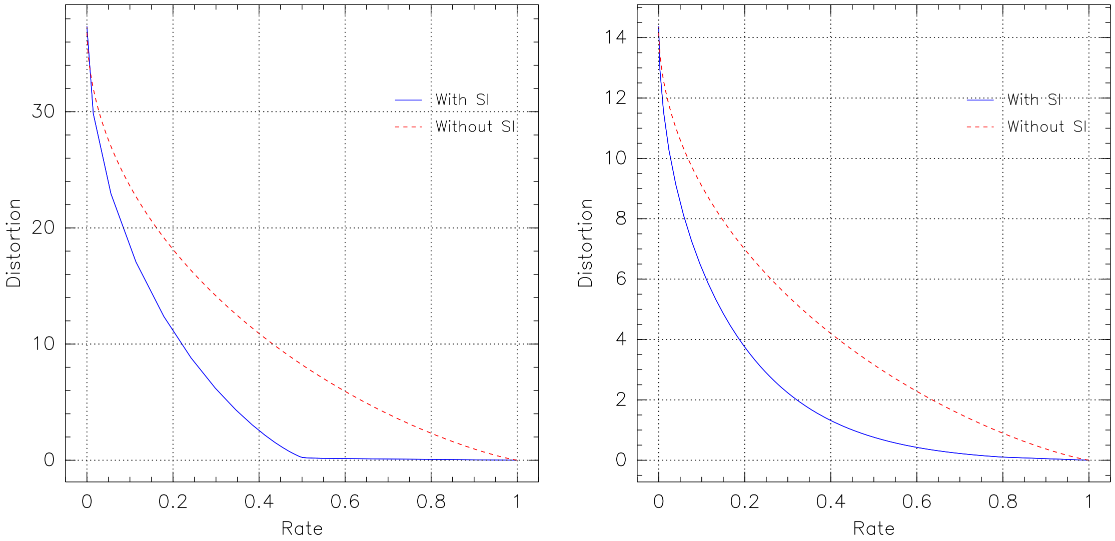


**Figure 2-6.** Rate-Distortion curves for a binary source with weighted Hamming distortion. The distortion for quantizing each source sample is measured via Hamming distortion times the weight $\exp(5 \cdot \mathsf{q})$. The solid curve represents the minimum possible Hamming distortion when side information specifying the weight is available at the encoder (or both at the encoder and decoder). The dashed curve represents the minimum distortion when no side information is available at the encoder. In the left plot, $\mathsf{q}$ is uniformly distributed over the pair $\{0, 1\}$ while in the right plot $\mathsf{q}$ is uniformly distributed over the interval $[0, 1]$.

### ■ 2.4.3 Erasure Distortions

Another scenario where the condition $I(\hat{\mathsf{s}}; \mathsf{q}) = 0$ is satisfied is for "erasure distortions" where $\mathsf{q}$ is binary and the distortion measure is of the form

$$d(s, \hat{s}; q) = \rho(s, \hat{s}) \cdot q \tag{2.14}$$

for some function $\rho(\cdot, \cdot)$.

**Theorem 2.** *For a distortion measure of the form in* (2.14)*, there exists an optimizing distribution for* $\hat{\mathsf{s}}$ *in* (2.4c) *with* $I(\hat{\mathsf{s}}; \mathsf{q}) = 0$ *and hence the rate-distortion function when* $\mathbf{q}$ *is known at only the encoder is the same when* $\mathbf{q}$ *is known at both encoder and decoder.*

*Proof.* Let $\hat{\mathsf{s}}^*$ be a distribution which optimizes (2.4d). Choose the new random variable $\hat{\mathsf{s}}^{**}$ to be the same as $\hat{\mathsf{s}}^*$ when $\mathsf{q} = 0$ and when $\mathsf{q} = 1$, let $\hat{\mathsf{s}}^{**}$ be independent of $\mathsf{s}$ with the same marginal distribution as when $\mathsf{q} = 0$:

$$p_{\hat{\mathsf{s}}^{**}|\mathsf{s},\mathsf{q}}(\hat{s}|s, q) = \begin{cases} p_{\hat{\mathsf{s}}^{*}|\mathsf{s},\mathsf{q}}(\hat{s}|s, q), & q = 0 \\ p_{\hat{\mathsf{s}}^{*}|\mathsf{q}}(\hat{s}|q = 0), & q = 1. \end{cases} \tag{2.15}$$

Both $\hat{\mathsf{s}}^*$ and $\hat{\mathsf{s}}^{**}$ have the same expected distortion since they only differ when $\mathsf{q} = 0$. Furthermore, by the data processing inequality

$$I(\hat{\mathsf{s}}^{**}; \mathsf{s}|\mathsf{q}) \leq I(\hat{\mathsf{s}}^{*}; \mathsf{s}|\mathsf{q}) \tag{2.16}$$

so $\hat{\mathsf{s}}^{**}$ also optimizes (2.4d). Finally, since $I(\hat{\mathsf{s}}^{**}; \mathsf{q}) = 0$, Proposition 2 is satisfied and we obtain the desired result. □

As shown in the examples of Section 2.2, erasure distortions may admit particularly simple quantizers to optimally use encoder side information.

## ■ 2.5 Asymptotically High-Resolution Quantization

In Proposition 2 of Section 2.4 we saw that when $\mathsf{w}$ is null, knowing $\mathsf{q}$ only at the encoder is as good as full knowledge of $\mathsf{q}$ when the codebook is independent of the distortion side information (*i.e.*, when $I(\hat{\mathsf{s}};\mathsf{q}) = 0$). In the high-resolution regime,[2] $\hat{\mathsf{s}}$ begins to closely approximate $\mathsf{s}$. Thus when $\mathsf{s}$ and $\mathsf{q}$ are independent we may intuitively expect that as $\hat{\mathsf{s}} \to \mathsf{s}$ we have $I(\hat{\mathsf{s}};\mathsf{q}) \to I(\mathsf{s};\mathsf{q}) = 0$. In this section, we rigorously justify this intuition and also considers some variations. For example, we consider the case when the signal side information, $\mathsf{w}$, which is statistically dependent on the source is present and we also consider the case where the source and distortion side information are not statistically independent. We begin by introducing some technical conditions, then state our main results, and close with an example.

### ■ 2.5.1 Technical Conditions

Our results require a "continuity of entropy" property which essentially states that

$$\mathsf{z} \to 0 \Rightarrow h(\mathsf{s}+\mathsf{z}|\mathsf{q},\mathsf{w}) \to h(\mathsf{s}|\mathsf{q},\mathsf{w}). \tag{2.17}$$

The desired continuity follows from [109] provided the source, distortion measure, and side information satisfy some technical conditions. These conditions are not particularly hard to satisfy; for example, any vector source, side information, and distortion measure, where

$$\exists\delta > 0, -\infty < E[||\mathbf{s}||^{\delta}|\mathbf{w} = \mathbf{w}] < \infty \quad \forall \mathbf{w} \tag{2.18a}$$

$$-\infty < h(\mathbf{s}|\mathbf{w} = \mathbf{w}) < \infty, \quad \forall \mathbf{w} \tag{2.18b}$$

$$d(\mathbf{s}, \hat{\mathbf{s}}; \mathbf{q}) = \alpha(\mathbf{q}) + \beta(\mathbf{q}) \cdot ||\mathbf{s} - \hat{\mathbf{s}}||^{\gamma(\mathbf{q})} \tag{2.18c}$$

satisfy the desired technical conditions provided $\alpha(\cdot)$, $\beta(\cdot)$, and $\gamma(\cdot)$ are non-negative functions. For more general scenarios we introduce the following definition to summarize the requirements from [109].

**Definition 1.** *We define a source $\mathsf{s}$, side information pair $(\mathsf{q},\mathsf{w})$, and difference distortion measure $d(s,\hat{s};q) = \rho(s-\hat{s};q)$ as admissible if the following conditions are satisfied:*

1. *the equations*

$$a(D,q)\int \exp[-s(D,q)\rho(s;q)]ds = 1 \tag{2.19a}$$

$$a(D,q)\int \rho(s;q)\exp[-s(D,q)\rho(s;q)]ds = D \tag{2.19b}$$

*have a unique pair of solutions $(a(D,q), s(D,q))$ for all $D > D_{\min}$ which are continuous functions of their arguments*

[2] Usually, the high-resolution limit is defined as when $D \to 0$, but it is sometimes useful to consider distortion measures with a constant penalty. Hence we assume it is possible to define a minimum distortion, $D_{\min}$, which can be approached arbitrarily closely as the rate increases and we define the high-resolution limit as $D \to D_{\min}$.

2. *$-\infty < h(\mathsf{s}|\mathsf{w} = w) < \infty$, for all $w$*

3. *For each value of $q$, there exists an auxiliary distortion measure $\delta(\cdot; q)$ where the equations*

$$a_\delta(D, q) \int \exp[-s_\delta(D, q)\delta(s; q)]ds = 1 \tag{2.20a}$$

$$a_\delta(D, q) \int \delta(s; q) \exp[-s_\delta(D, q)\delta(s; q)]ds = D \tag{2.20b}$$

*have a unique pair of solutions $(a_\delta(D, q), s_\delta(D, q))$ for all $D > D_{\min}$ which are continuous functions of their arguments*

4. *The distribution $\mathsf{z}_\mathsf{q}$ which maximize $h(\mathsf{z}_\mathsf{q}|\mathsf{q})$ subject to the constraint $E[\rho(\mathsf{z}_\mathsf{q}; \mathsf{q})] \leq D$ has the property that*

$$\lim_{D \to D_{\min}} \mathsf{z}_q \to 0 \text{ in distribution } \forall q \tag{2.21a}$$

$$\lim_{D \to D_{\min}} E[\delta(\mathsf{s} + \mathsf{z}_\mathsf{q}, \mathsf{q})|\mathsf{q} = q] = E[\delta(\mathsf{s}, \mathsf{q})|\mathsf{q} = q] \quad \forall q. \tag{2.21b}$$

## ■ 2.5.2 Equivalence Theorems

Our main results for continuous sources consist of various theorems describing when different types of side information knowledge are equivalent (*e.g.*, knowing **q** at the encoder is as good as knowing it at both when **q** and **w** are independent). The theorems stated below are proved in Appendix B.3 and summarized in Fig. 2-7. Essentially, we show that the sixteen possible information configurations can be reduced to the four shown in Fig. 2-7. Specifically, to determine the performance of a given side information configuration one can essentially ask two questions: "Does the encoder have $\mathsf{q}$?" and "Does the decoder have $\mathsf{w}$?". A negative answer to either question incurs some penalty relative to the case with all side information known everywhere. In contrast, a positive answer to both questions incurs asymptotically no rate-loss relative to complete side information.

| | Decoder missing **w** | | | Decoder has **w** | | |
|---|---|---|---|---|---|---|
| Encoder Missing **q** | $R$[Q-DEC-W-ENC] | $\overset{\text{Th. 4 (I)}}{\Longleftrightarrow}$ | $R$[Q-DEC-W-NONE] | $R$[Q-DEC-W-BOTH] | $\overset{\text{Th. 5 (S)}}{\Longleftrightarrow}$ | $R$[Q-NONE-W-BOTH] |
| | $\Updownarrow$ Th. 5 (S+I) | | $\Updownarrow$ Th. 5 (S+I) | $\Updownarrow$ Th. 6 (H+S+I) | | $\Updownarrow$ Th. 6 (H+S+I) |
| | $R$[Q-NONE-W-ENC] | $\overset{\text{Th. 4 (I)}}{\Longleftrightarrow}$ | $R$[Q-NONE-W-NONE] | $R$[Q-DEC-W-DEC] | $\overset{\text{Th. 5 (S)}}{\Longleftrightarrow}$ | $R$[Q-NONE-W-DEC] |
| Encoder has **q** | $R$[Q-ENC-W-ENC] | $\overset{\text{Th. 4}}{\Longleftrightarrow}$ | $R$[Q-ENC-W-NONE] | $R$[Q-ENC-W-DEC] | $\overset{\text{Th. 6 (H)}}{\Longleftrightarrow}$ | $R$[Q-ENC-W-BOTH] |
| | $\Updownarrow$ Th. 6 (H+I) | | $\Updownarrow$ Th. 6 (H+I) | $\Updownarrow$ Th. 6 (H) | | $\Updownarrow$ Th. 6 (H) |
| | $R$[Q-BOTH-W-ENC] | $\overset{\text{Th. 4}}{\Longleftrightarrow}$ | $R$[Q-BOTH-W-NONE] | $R$[Q-BOTH-W-DEC] | $\overset{\text{Th. 6 (H)}}{\Longleftrightarrow}$ | $R$[Q-BOTH-W-BOTH] |

**Figure 2-7.** Summary of main results for continuous sources. Arrows indicate which theorems demonstrate equality between various rate-distortion functions and list the assumptions required (H = high-resolution, I = **q** and **w** independent, S = scaled difference distortion).

We begin by generalizing previous results about no rate-loss for the Wyner-Ziv problem in the high-resolution limit [193] [110] to show there is no rate-loss when **q** is known at the encoder and **w** is known at the decoder. This results suggests that there is a natural division of side information between the encoder and decoder (at least asymptotically). Specifically, distortion side information should be sent to the encoder and signal side information should be sent to the decoder.

**Theorem 3.** *For any admissible source, side information, and difference distortion measure satisfying Definition 1,* **q** *and* **w** *can be divided between the encoder and decoder with no asymptotic penalty,* i.e.,

$$\lim_{D \to D_{\min}} R[\text{Q-ENC-W-DEC}](D) - R[\text{Q-BOTH-W-BOTH}](D) = 0. \tag{2.22}$$

The next two theorems generalize Berger's result that signal side information is useless when known only at the encoder [19] to show when **w** and **q** are useless at the encoder and decoder respectively. These results suggest that deviating from the natural division of Theorem 3 and providing side information in the wrong place makes that side information useless (at least in terms of the rate-distortion function).

**Theorem 4.** *Let* **q** *and* **w** *be independent*[3] *and consider a difference distortion measure of the form $d(s - \hat{s}; q)$. Then* **w** *provides no benefit when known only at the encoder,* i.e.,

$$R[\text{Q-*-W-ENC}](D) = R[\text{Q-*-W-NONE}](D) \tag{2.23}$$

*where the wildcard "*" may be replaced with an element from* $\{\text{ENC}, \text{DEC}, \text{BOTH}, \text{NONE}\}$ *(both *'s must be replaced with the same element).*

**Theorem 5.** *Let* **q** *and* **w** *be independent*[4] *and consider a scaled distortion measure of the form $d(s, \hat{s}; q) = d_0(q) d_1(s, \hat{s})$. Then* **q** *provides no benefit when known only at the decoder,* i.e.,

$$R[\text{Q-DEC-W-*}](D) = R[\text{Q-NONE-W-*}](D) \tag{2.24}$$

*where the wildcard "*" may be replaced with an element from* $\{\text{ENC}, \text{DEC}, \text{BOTH}, \text{NONE}\}$ *(both *'s must be replaced with the same element).*

Using the previous results, we can generalize Theorem 3 to show that regardless of where **w** (respectively **q**) is known, knowing **q** (**w**) only at the encoder (decoder) is sufficient in the high-resolution limit. This result essentially suggests that even if the ideal of providing **q** to the encoder and **w** to the decoder suggested by Theorem 3 is impossible, it is still useful to follow this ideal as much as possible.

**Theorem 6.** *Let* **q** *and* **w** *be independent.*[5] *For any source and scaled*[6] *difference distortion measure $d(s, \hat{s}; q) = d_0(q) \cdot d_1(s - \hat{s})$ satisfying the conditions in Definition 1,* **q** *(respectively,* **w***) is asymptotically only required at the encoder (respectively, at the decoder),* i.e.,

$$\lim_{D \to D_{\min}} R[\text{Q-ENC-W-*}](D) - R[\text{Q-BOTH-W-*}](D) = 0 \tag{2.25a}$$

$$\lim_{D \to D_{\min}} R[\text{Q-*-W-DEC}](D) - R[\text{Q-*-W-BOTH}](D) = 0 \tag{2.25b}$$

*where the wildcard "*" may be replaced with an element from* $\{\text{ENC}, \text{DEC}, \text{BOTH}, \text{NONE}\}$ *(both *'s must be replaced with the same element).*

[3] Independence is only required when $* \in \{\text{DEC}, \text{NONE}\}$; if $* \in \{\text{ENC}, \text{BOTH}\}$, the theorem holds without this condition.

[4] Independence is only required when $* \in \{\text{ENC}, \text{NONE}\}$; if $* \in \{\text{DEC}, \text{BOTH}\}$, the theorem holds without this condition.

[5] Independence is only required when $* \in \{\text{ENC}, \text{NONE}\}$ in (2.25a) or when $* \in \{\text{DEC}, \text{NONE}\}$ in (2.25b). For $* \in \{\text{DEC}, \text{BOTH}\}$ in (2.25a) or $* \in \{\text{ENC}, \text{BOTH}\}$ in (2.25b) the theorem holds without this condition.

[6] The scaled form of the distortion measure is only required when $* \in \{\text{DEC}, \text{NONE}\}$ in (2.25b). When $* \in \{\text{ENC}, \text{BOTH}\}$, the theorem only requires a difference distortion measure of the form $d(s, \hat{s}; q) = d'(s - \hat{s}; q)$.

### ■ 2.5.3 Penalty Theorems

We can compute the asymptotic penalty for not knowing the signal side information $\mathsf{w}$ at the decoder.

**Theorem 7.** *Let $\mathsf{q}$ and $\mathsf{w}$ be independent. For any source and scaled difference distortion measure $d(s,\hat{s};q) = d_0(q) \cdot d_1(s-\hat{s})$ satisfying the conditions in Definition 1, the penalty for not knowing $\mathsf{w}$ at the decoder is*

$$\lim_{D \to D_{\min}} R[\textsc{q-*-w-\{enc-or-none\}}](D) - R[\textsc{q-*-w-\{dec-or-both\}}](D) = I(\mathsf{s};\mathsf{w}) \tag{2.26}$$

*where the wildcard "*" may be replaced with an element from $\{\textsc{enc}, \textsc{dec}, \textsc{both}, \textsc{none}\}$ (all *'s must be replaced with the same element).*

Theorem 7 also gives us insight about distortion side information which is not independent of the source. Specifically, imagine that the side information $\mathsf{z}$ affects the distortion via $d(s,\hat{s};z) = d_0(z) \cdot d_1(s-\hat{s})$ and furthermore, $\mathsf{z}$ is correlated with the source. What is the penalty for knowing $\mathsf{z}$ only at the encoder? To answer this question, we can decompose $\mathsf{z}$ into our framework by setting $\mathsf{q} = \mathsf{z}$ and $\mathsf{w} = \mathsf{z}$. This pair of $\mathsf{q}$ and $\mathsf{w}$ trivially satisfies the conditions of Theorem 7. Therefore, by substituting $\mathsf{z}$ into Theorem 7, we see that the asymptotic penalty for knowing general side information only at the encoder is exactly the degree to which the source and distortion side information are related as measured by mutual information:

**Corollary 1.** *For any source and scaled difference distortion measure $d(s,\hat{s};z) = d_0(z) \cdot d_1(s-\hat{s})$ satisfying the conditions in Definition 1, the penalty for knowing general side information, $\mathsf{z}$, only at the encoder is*

$$\lim_{D \to D_{\min}} R[\textsc{z-enc}](D) - R[\textsc{z-both}](D) = I(\mathsf{s};\mathsf{z}). \tag{2.27}$$

Finally, we can compute the asymptotic penalty for not knowing the distortion side information $\mathsf{q}$ at the encoder.

**Theorem 8.** *Let $\mathsf{q}$ and $\mathsf{w}$ be independent. For any source taking values in the $k$-dimensional real vector space and a scaled norm-based distortion measure $d(\mathbf{s},\hat{\mathbf{s}};\mathbf{q}) = \mathbf{q} \cdot ||\mathbf{s}-\hat{\mathbf{s}}||^r$ satisfying the conditions in Definition 1, the penalty for not knowing $\mathsf{q}$ at the encoder is*

$$\lim_{D \to D_{\min}} R[\textsc{q-\{dec-or-none\}-w-*}](D) - R[\textsc{q-\{enc-or-both\}-w-*}](D) = \frac{k}{r} E\left[\ln \frac{E[\mathsf{q}]}{\mathsf{q}}\right] \tag{2.28}$$

*where the wildcard "*" may be replaced with an element from $\{\textsc{enc}, \textsc{dec}, \textsc{both}, \textsc{none}\}$ (both *'s must be replaced with the same element).*

A similar result which essentially compares the asymptotic difference between

$$R[\textsc{q-dec-w-dec}](D) \text{ and } R[\textsc{q-both-w-both}](D)$$

for non-independent $\mathsf{q}$ and $\mathsf{w}$ with squared norm distortion is derived in [110]. Thus, as with Corollary 1 and [110], Theorem 8 can be interpreted as saying that the asymptotic penalty for knowing general side information only at the decoder can be quantified by the degree to which the distortion and the side information are related as measured by (2.28).

According to Jensen's inequality, this rate gap is always greater than or equal to zero with equality if and only if $q$ is a constant with probability 1. Furthermore, the rate gap is scale invariant in the sense that it does not change when $q$ is multiplied by any positive constant.

In Table 2.1, we evaluate the high-resolution rate penalty for a number of possible distributions for the side-information. Note that for all of these side information distributions (except the uniform and exponential distributions), the rate penalty can be made arbitrarily large by choosing the appropriate shape parameter to place more probability near $q = 0$ or $q = \infty$. In the former case (LogNormal, Gamma, or Pathological $q$), the large rate-loss occurs because when $q \approx 0$, the informed encoder can transmit almost zero rate while the uninformed encoder must transmit a large rate to achieve high resolution. In the latter case (Pareto or Cauchy $q$), the large rate-loss is caused by the heavy tails of the distribution for $q$. Specifically, even though $q$ is big only very rarely, it is the rare samples of large $q$ which dominate the moments. Thus an informed encoder can describe the source extremely accurately during the rare occasions when $q$ is large, while an uninformed encoder must always spend a large rate to obtain a low average distortion.

**Table 2.1.** Asymptotic rate-penalty in nats. Euler's constant is denoted by $\gamma$.
The rate penalties below are for not knowing distortion side information $q$ at the encoder when distortion is measured via $d(s, \hat{s}; q) = q(s - \hat{s})^2$. (Multiply penalties in nats by $1/\ln 2 \approx 1.44$ to convert to bits).

| Distribution Name | Density for $q$ | Rate Gap in nats |
|---|---|---|
| Exponential | $\tau \exp(-q\tau)$ | $-\frac{1}{2}\ln\gamma \approx 0.2748$ |
| Uniform | $1_{q\in[0,1]}$ | $\frac{1}{2}(1-\ln 2) \approx 0.1534$ |
| Lognormal | $\frac{1}{q\sqrt{2\pi Q^2}}\exp\left[-\frac{(\ln q-M)^2}{2Q^2}\right]$ | $\frac{Q^2}{4}$ |
| Pareto | $\frac{a^b}{q^{a+1}}, q \geq b > 0, a > 1$ | $\frac{1}{2}\left[\ln\frac{a}{a-1} - 1/a\right]$ |
| Gamma | $\frac{b(bq)^{a-1}\exp(-bq)}{\Gamma(a)}$ | $\frac{1}{2}\left\{\ln a - \frac{d}{dx}[\ln\Gamma(x)]_{x=a}\right\} \approx \frac{1}{2a}$ |
| Pathological | $(1-\epsilon)\delta(q-\epsilon) + \epsilon\delta(q-1/\epsilon)$ | $\frac{1}{2}\ln(1+\epsilon-\epsilon^2) - \frac{1-2\epsilon}{2}\ln\epsilon \approx \frac{1}{2}\ln\frac{1}{\epsilon}$ |
| Positive Cauchy | $\frac{2/\pi}{1+q^2}, q \geq 0$ | $\infty$ |

Finally, note that all but one of these distributions would require infinite rate to losslessly communicate the side information. This suggests that the gains in using distortion side information are *not* obtained by exactly describing the side information to the receiver.

## ■ 2.6 Finite Rate Bounds For Quadratic Distortions

So far we have shown that in the high-resolution limit, knowing $\mathbf{q}$ at the encoder is sufficient. Our main analytical tool was the additive test-channel distribution $\hat{s} = s + z_q$ where the additive noise in the test channel depends on the distortion side information. Evidently, additive test-channels of this type are asymptotically optimal. To investigate the rate-loss at finite resolutions we develop two results for scaled quadratic distortion measures. We also briefly mention how these results can be generalized to other distortion measures.

### 2.6.1 A Medium Resolution Bound

**Theorem 9.** *Consider a scaled quadratic distortion measure of the form $d(s, \hat{s}; q) = q \cdot (s - \hat{s})^2$ with $q \geq q_{\min} > 0$. Then the maximum rate-gap at distortion $D$ is bounded by*

$$R[\text{Q-ENC-W-DEC}](D) - R[\text{Q-BOTH-W-BOTH}](D) \leq \frac{J(\mathsf{s}|\mathsf{w})}{2} \cdot \min\left[1, \frac{D}{q_{\min}}\right] \tag{2.29}$$

*where $J(\mathsf{s}|\mathsf{w})$ is the Fisher Information in estimating a non-random parameter $\tau$ from $\tau + \mathsf{s}$ conditioned on knowing $\mathsf{w}$. Specifically,*

$$J(\mathsf{s}|\mathsf{w}) \triangleq \int p_{\mathsf{w}}(w) \left\{ \int p_{\mathsf{s}|\mathsf{w}}(s|w) \left[\frac{\delta}{\delta s} \log p_{\mathsf{s}|\mathsf{w}}(s|w)\right]^2 ds \right\} dw. \tag{2.30}$$

Similar bounds can be developed with other distortion measures provided that $D/q_{\min}$ is replaced with a quantity proportional to the variance of the quantization error. See the remark after the proof of Theorem 9 in the appendix for details.

One may wonder why Fisher Information appears in the above rate-loss bound. After all, Fisher Information is most commonly used to lower bound the error in estimating a parameter via its use in the Cramer-Rao bound. What does Fisher Information have to do with source coding?

To answer this question, recall that our bounds are all developed by using an additive test-channel distribution of the form $\hat{\mathsf{s}} = \mathsf{s} + \mathsf{z}$. Thus, a clever source decoder could treat each source sample $\mathsf{s}[i]$ as a parameter to be "estimated" from the quantized representation $\hat{\mathsf{s}}[i]$. If an efficient estimator exists, this procedure could potentially reduce the distortion by the reciprocal of the Fisher Information. But if the distortion can be reduced in this manner without affecting the rate, then the additive test-channel distribution must be sub-optimal and a rate gap must exist.

So the bound in Theorem 9 essentially measures the rate gap by measuring how much our additive test-channel distribution could be improved if an efficient estimator existed for $\mathsf{s}$ given $\hat{\mathsf{s}}$. This bound will tend to be good when an efficient estimator does exist and poor otherwise. For example, if $\mathsf{s}$ is Gaussian with unit-variance conditioned on $\mathsf{w}$, then the Fisher Information term in (2.29) evaluates to one and the worst-case rate-loss is at most half a bit at maximum distortion. This corresponds to the half-bit bound on the rate-loss for the pure Wyner-Ziv problem derived in [193]. But if $\mathsf{s}$ is discontinuous (*e.g.*, if $\mathsf{s}$ is uniform), then no efficient estimator exists and the bound in (2.29) is poor.

As an aside, we note that the proof of Theorem 9 does not require any regularity conditions. Hence, if the Fisher Information of the source is finite, it can be immediately applied without the need to check whether the source is admissible according to Definition 1.

### 2.6.2 A Low Resolution Bound

While the Fisher Information bound from (2.29) can be used at low resolutions, it can be quite poor if the source is not smooth. Therefore, we derive the following bound on the rate-loss which is independent of the distortion level and hence most useful at low resolution.

**Theorem 10.** *Consider a scaled quadratic distortion measure of the form $d(s, \hat{s}; q) = q \cdot (s - \hat{s})^2$ and denote the minimum/maximum conditional variance of the source by*

$$\sigma^2_{\text{min}} = \min_{w} \operatorname{Var}[\mathsf{s}|\mathsf{w} = w] \tag{2.31a}$$

$$\sigma^2_{\text{max}} = \max_{w} \operatorname{Var}[\mathsf{s}|\mathsf{w} = w]. \tag{2.31b}$$

*Then the gap in Theorem 3 is at most the conditional relative entropy of the source from a Gaussian distribution plus a term depending on the range of the conditional variance:*

$$R[\text{Q-ENC-W-DEC}](D) - R[\text{Q-BOTH-W-BOTH}](D) \leq D(\mathsf{s}|\mathsf{w}||\mathcal{N}(\operatorname{Var}[\mathsf{s}])) + \frac{1}{2}\log\left(1 + \frac{\sigma^2_{\text{max}}}{\sigma^2_{\text{min}}}\right) \tag{2.32}$$

*where $\mathcal{N}(t)$ represents a Gaussian random variable with mean zero and variance $t$.*

Similar bounds can be developed for other distortion measures as discussed after the proof of Theorem 10 in the appendix.

Consider the familiar Wyner-Ziv scenario where the signal side information is a noisy observation of the source. Specifically, let $\mathsf{w} = \mathsf{s} + \mathsf{v}$ where $\mathsf{v}$ is independent of $\mathsf{s}$. In this case, the conditional variance is constant and (2.32) becomes

$$D(\mathsf{s}|\mathsf{w}||\mathcal{N}(\operatorname{Var}[\mathsf{s}])) + \frac{1}{2}\log 2 \tag{2.33}$$

and the rate-loss is at most half a bit plus the deviation from Gaussianity of the source.

If $\mathsf{s}$ is Gaussian when conditioned on $\mathsf{w} = w$, then the rate-loss is again seen to be at most half a bit as in [193]. In contrast to [193] which is independent of the source, however, both our bounds in (2.29) and (2.32) depend on the source distribution. Hence, we conjecture that our bounds are loose. In particular, for a discrete source, the worst case rate loss is at most $H(\mathsf{s}|\mathsf{w})$, but this is not captured by our results since both bounds become infinity. Techniques from [60, 104, 193] may yield tighter bounds.

### 2.6.3 A Finite Rate Gaussian Example

To gain some idea of when the asymptotic results take effect, we consider a finite rate Gaussian example. Specifically, let the source consist of a sequence of Gaussian random variables with mean zero and variance 1 and consider distortion side information with $\Pr[\mathsf{q} = 1] = 0.6$, $\Pr[\mathsf{q} = 10] = 0.4$, and distortion measure $d(s, \hat{s}; q) = q \cdot (s - \hat{s})^2$.

The case without side information is equivalent to quantizing a Gaussian random variable with distortion measure $4.6(\mathsf{s} - \hat{\mathsf{s}})^2$ and thus the rate-distortion function is

$$R[\text{Q-NONE-W-NONE}](D) = \begin{cases} 0, & D \geq 4.6 \\ \frac{1}{2}\ln\frac{4.6}{D}, & D \leq 4.6. \end{cases} \tag{2.34}$$

To determine $R[\text{Q-BOTH-W-NONE}](D)$ we must set up a constrained optimization as we did for the binary-Hamming scenario in Appendix B.2. This optimization results in a "water-pouring" bit allocation which uses more bits to quantize the source when $\mathsf{q} = 10$ than when $\mathsf{q} = 1$. Specifically, the optimal test-channel is a Gaussian distribution where both the mean and the variance depend on $\mathsf{q}$ and thus $\hat{\mathsf{s}}$ has a Gaussian mixture distribution. Going through the details of the constrained optimization yields

$$R[\text{Q-BOTH-W-NONE}](D) = \begin{cases} 0, & 4.6 \le D \\ \frac{0.4}{2}\ln\frac{4}{(D-.6)}, & D^* \le D \le 4.6 \\ \frac{0.4}{2}\ln\frac{10}{D} + \frac{0.6}{2}\ln\frac{1}{D}, & D \le D^* \end{cases} \tag{2.35}$$

for some appropriate threshold $D^*$. Evaluating (2.28) for this case indicates that the rate-gap between (2.34) and (2.35) goes to $0.5 \cdot (\ln 4.6 - 0.4 \ln 10) \approx 0.3$ nats $\approx 0.43$ bits.

Computing $R[\text{Q-ENC-W-NONE}](D)$ analytically seems difficult. Thus, when distortion side information is only available at the encoder we obtain a numerical upper bound on the rate by using the same codebook distribution as when $q$ is known at both encoder and decoder. This yields a rate penalty of $I(\hat{s}; q)$.[7] We can obtain a simple analytic bound from Theorem 9. Specifically, evaluating (2.29) yields that the rate penalty is at most $(1/2) \cdot \min[1, D]$.

In Fig. 2-8 we evaluate these rate-distortion trade-offs. We see that at zero rate, all three rate-distortion functions have the same distortion since no bits are available for quantization. Furthermore, we see that the Fisher Information bound is loose at zero rate. As the rate increases, the system with full distortion side-information does best because it uses the few available bits to represent only the important source samples with $q = 10$. The decoder reconstructs these source samples from the compressed data and reconstructs the less important samples to zero (the mean of $s$). In this regime, the system with distortion side information at the encoder also more accurately quantizes the important source samples. But since the decoder does not know $q$, it does not know which samples of $\hat{s}$ to reconstruct to zero. Thus the system with $q$ available at the encoder performs worse than the one with $q$ at both encoder and decoder but better than the system without side information. As the rate increases further, both systems with distortion side information quantize source samples with both $q = 1$ and $q = 10$. Thus the codebook distribution for $\hat{s}$ goes from a Gaussian mixture to become more and more Gaussian and the rate-loss for the system with only encoder side information goes to zero. Finally, we note that even at the modest distortion of $-5$ dB, the asymptotic effects promised by our theorems have already taken effect.

## ■ 2.7 Discussion

In this chapter, we introduced the notion of distortion side information which does not directly depend on the source but instead affects the distortion measure. Furthermore, we showed that if general side information can be decomposed into such a distortion dependent component and a signal dependent component then under certain conditions the former is required only at the encoder and the latter is required only at the decoder. In this section, we discuss some implications and applications of these ideas.

### ■ 2.7.1 Applications to Sensors and Sensor Networks

There is a growing interest in sensor networks where multiple nodes with potentially correlated observations cooperate to sense the environment. A variety of researchers have already

[7] Actually, since the rate distortion function is convex, we take the lower convex envelope of the curve resulting from the optimal test-channel distribution.

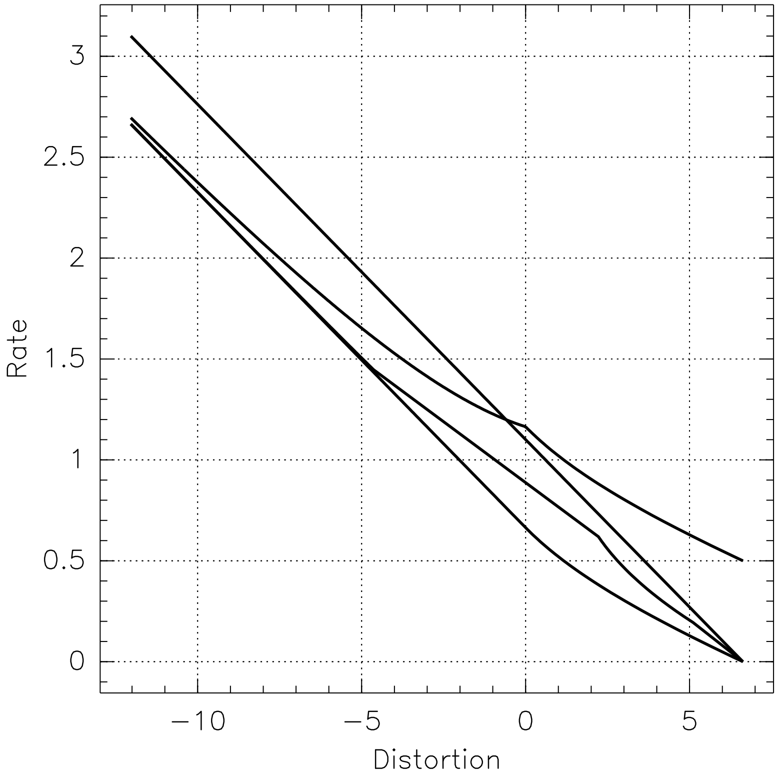


**Figure 2-8.** Rate-distortion curves for quantizing a Gaussian source $s$ with distortion $q(s - \hat{s})^2$ where the side information $q$ is 1 with probability 0.6 or 10 with probability 0.4. From bottom to top on the right the curves correspond to the rate required when both encoder and decoder know $q$, a numerically computed upper bound to the rate when only the encoder knows $q$, the rate when neither encoder nor decoder know $q$, and the Fisher Information upper bound from Theorem 9 for when only the encoder knows $q$.

demonstrated the advantages of distributed source coding in efficiently using signal side information in such networks [149, 146, 136, 175, 21]. We believe, however, that the possible applications of distributed coding with distortion side information are equally compelling.

First, many sensors naturally receive distortion side information in addition to the observed signal. For example, sensors which perform simple filtering or averaging of the incoming signal can use the variance as an indicator of reliability. Similarly, sensors can observe the background level of light, sound, *etc.*, to obtain an estimate of the noise-floor in the absence of a signal of interest. Since the level of this noise-floor may change over time, such background noise readings can be used as an estimate of signal reliability.

Second, many sensors must be periodically re-calibrated either by manual intervention, with automatic self-calibration routines, or by communicating with other sensors. The length of time since the last calibration or an estimate of the calibration accuracy are thus natural measures of reliability for the observations. For example, many systems are designed to record data at a roughly constant average amplitude. This is often accomplished by using automatic gain control (AGC) to compensate for attenuation due to distance, weather, or obstacles. Since thermal noise in the sensor front-end is usually independent of such attenuations, the AGC level can be used as a simple indicator of the signal-to-noise ratio of an observed signal.

In more sophisticated systems, intermediate nodes or fusion centers may collect data and perform various levels of processing. These fusion centers can often form preliminary estimates of the value of different components of the data. These estimates can the be used as distortion side information in deciding the accuracy used to further communicate or store the processed data.

In all these cases, our results suggest that trying to communicate distortion side information along with the signal is often not the right way to design a system. Instead, compression schemes should be designed such that the decoder is in some sense invariant to the side information. Of course, the encoder must still be able to more accurately represent the important or reliable data. As illustrated by the examples in Section 2.2, various transforms may play a role in achieving this goal.

### ■ 2.7.2 Richer Distortion Models and Perceptual Coding

Notion of distortion side information may be particularly useful in the design of perceptual coders. Specifically, it is well-known that mean-square error is at best a poor representative of how humans experience distortion. Hence perceptual coding systems use features of the human visual system (HVS) or human auditory system (HAS) to achieve low subjective distortion even when the mean-square distortion is quite large. Unfortunately, creating such a perceptual coder often requires the designer to be an expert both in human physiology, as well as quantizer design. This difficulty may be one of the reasons why information theory has sometimes had less impact on compression standards than on communication standards.

Using the abstraction of distortion side information to represent such perceptual effects, however, may help overcome this barrier. For example, physiological experts could focus on producing a distortion model which incorporates perceptual effects by determining a value $q_i$ which scales the quadratic distortion between the $i$th source sample and its reconstruction. Quantization experts could focus on designing compression systems operating with distortion side information. Then various perceptual models could be quickly and easily combined with quantizer designs to find the best combination. This modular design would allow a good quantization system to be used in a variety of application domains simply by changing the model for the distortion side information. Similarly, it would allow a finely tuned perceptual model to be used in many types of quantizer designs.

The theoretical justification for using distortion side information to design modular quantizers is Theorem 3. This theorem can be interpreted as saying that even if the perceptual model which produces the distortion weighting $q_i$ is a complicated function of the source, the decoder does not need to know how the perceptual model works. Instead, it is sufficient for the decoder simply to know the statistics of $q$ provided that $q$ is available to the encoder.

Finally, it may be somewhat premature to advocate a particular design for perceptual coders based on primarily on rate-distortion results. At a minimum, however, we point out that if a perceptual distortion side information model can be constructed it can at least be used to find a bound on the minimum possible bit rate at a given distortion. Having such a performance benchmark to strive for can be a powerful force in inspiring system designer to search for new innovations.

### ■ 2.7.3 Decomposing Side Information Into q and w

Our problem model of Section 2.3 requires that the side information can be decomposed into distortion side information and signal side information. In many scenarios, this may be a natural view of the compression task. In problems where this is not the case, however, we briefly discuss how general side information may be decomposed into the pair $(\mathbf{q}, \mathbf{w})$ required by our theorems.

First, notice that the formulation in (2.1) is completely general in the sense that any side information $z$ taking values in the set $\mathcal{Z}$ can be trivially decomposed into $(q, w)$ by setting $(q, w) = (z, z)$ with $(\mathcal{Q}, \mathcal{W}) = (\mathcal{Z}, \mathcal{Z})$. Of course, this makes most of our results uninteresting. One systematic procedure for potentially improving this decomposition is as follows. First, replace each pair of values $(q_0, q_1)$ where $d(\cdot, \cdot; q_0) = d(\cdot, \cdot; q_1)$ with a new value $q'$ and adjust $\mathcal{Q}$ accordingly. Next, replace each pair of values $(w_0, w_1)$ where $p_{s|w}(\cdot|w_0) = p_{s|w}(\cdot|w_1)$ and $p_{q|w}(\cdot|w_0) = p_{q|w}(\cdot|w_1)$ with a new value $w'$ and adjust $\mathcal{W}$ accordingly. If the resulting $\mathcal{Q}$ and $\mathcal{W}$ are smaller than $\mathcal{Z}$, then our results become non-trivial.

As with the problem of determining a minimal sufficient statistics, there are many potential decompositions and different notions of good decompositions may be appropriate in different applications. For example, minimizing the cardinality of $\mathcal{Q}$, $\mathcal{W}$, or both, might be useful in simplifying quantizer design or related tasks. Alternatively, minimizing $I(q; w)$, $H(q)$, $H(w)$, or similar information measures may be useful if $\mathbf{q}/\mathbf{w}$ must be communicated to the encoder/decoder by a third party.

Essentially, all that a side information decomposition, requires for our results to hold are two Markov conditions. First, we need the Markov condition $\mathbf{q} \leftrightarrow \mathbf{w} \leftrightarrow \mathbf{s}$. Second, if we view the distortion as a random variable $d(\mathbf{s}, \hat{\mathbf{s}})$ which can depend on the side information, we need the Markov condition $d(\mathbf{s}, \hat{\mathbf{s}}) \leftrightarrow \mathbf{q}, \mathbf{s}, \hat{\mathbf{s}} \leftrightarrow \mathbf{w}$. In some scenarios, natural decompositions exist which satisfies these conditions, but finding a general procedure to obtain the best such decomposition is a topic for future work.

## 2.8 Concluding Remarks

Our analysis indicates that side information which affects the distortion measure can provide significant benefits in source coding. Perhaps, our most surprising result is that in a number of cases, (*e.g.*, sources uniformly distributed over a group, or in the high-resolution limit) side information at the encoder is just as good as side information known at both encoder and decoder. Furthermore, this "separation theorem" can be composed with the previously known result that having signal side information at the decoder is often as good as having it both encoder and decoder (*e.g.*, in the high-resolution limit). Our main results regarding when knowing a given type of side information at one place is as good as knowing it at another place are summarized in Fig. 2-7. Also, we computed the rate-loss for lacking a particular type of side information in a specific place. These penalty theorems show that lacking the proper side information can produce arbitrarily large degradations in performance. Taken together, we believe these results suggest that distortion side information is a useful source coding paradigm.

### 2.8.1 Possibilities For Future Work

Many practical, theoretical, and modeling issues remain to be explored in the study of distortion side information.

Probably the most pressing modeling issue is the investigation and development of useful distortion side information measures. A standard model (or class of models) for distortion side information in various applications would be invaluable in addressing semantic aspects of distortion ignored by cruder measures such as squared or absolute error. Developing such models may be a lengthy, expensive, and even controversial task. Similar difficulties are encountered in developing channel models for many wired and wireless communication applications, yet the value of such models makes them an active area of research.

Besides modeling questions, an interesting practical issue is the construction of low complexity quantizers which can approach the theoretical advantages of distortion side information in real systems. Some possibilities include investigating low dimensional quantizers via Lloyd-Max algorithms, lattice quantizers or companding. High dimensional constructions based on sophisticated codes may also prove useful if low complexity encoding and decoding algorithms can be constructed.

On the theoretical side, some interesting questions include the benefits of distortion side information in scalable or distributed source coding problems, the theory of distortion side information for non-difference distortion measures, and multiple description coding with distortion side information.

Chapter 3

# Low Complexity Quantizers For Distortion Side Information

## ■ 3.1 Introduction

Most classical work on lossy data compression and source coding focuses on encoding a source into a compressed representation such that compressed version is "close" to the original as measured by some distortion function. Choosing an accurate distortion measure is critical. Unfortunately, however, sophisticated distortion measures are often difficult to work with in analysis and design. Thus one common model is to measure the distortion between a length $n$ source sequence, $\mathsf{s}\left[{}^{n}_{1}\right]$, and its quantized reconstruction, $\hat{\mathsf{s}}\left[{}^{n}_{1}\right]$, by averaging a single-letter distortion measure, *i.e.*

$$d(\mathsf{s}\left[{}^{n}_{1}\right], \hat{\mathsf{s}}\left[{}^{n}_{1}\right]) = \frac{1}{n}\sum_{i=1}^{n} d(\mathsf{s}[i], \hat{\mathsf{s}}[i]). \tag{3.1}$$

Single-letter distortion measures have led to a number of theoretical insights and practical advances, but they seem to miss a key feature of many problems. For example, consider a sensor application with the observations $\mathsf{s}\left[{}^{n}_{1}\right]$. Furthermore, imagine that each observation is made with a different reliability or signal-to-noise ratio (SNR). When $\mathsf{s}\left[{}^{n}_{1}\right]$ is quantized to $\hat{\mathsf{s}}\left[{}^{n}_{1}\right]$, our intuition suggests that the high reliability or high SNR observations should be treated differently than the low reliability or low SNR ones. As another example, consider the encoding of speech, audio, or video. It is well-known that due to perceptual effects, certain components of such signal may be more or less sensitive to distortion due to masking effects or context [85]. For example errors in audio samples following a loud sound, or errors in pixels spatially or temporally near bright spots are perceptually less relevant. Similarly, accurately preserving certain edges or textures in an image or human voices in audio may be more important than preserving background patterns/sounds.

To model some of these effects in, we introduced the idea of distortion side information in Chapter 2. Specifically, we considered quantizing a source $\mathsf{s}\left[{}^{n}_{1}\right]$ into $\hat{\mathsf{s}}\left[{}^{n}_{1}\right]$ subject to a distortion measure which depends on the side information $\mathsf{q}\left[{}^{n}_{1}\right]$ via

$$d(\mathsf{s}\left[{}^{n}_{1}\right], \hat{\mathsf{s}}\left[{}^{n}_{1}\right]; \mathsf{q}\left[{}^{n}_{1}\right]) = \frac{1}{n}\sum_{i=1}^{n} d(\mathsf{s}[i], \hat{\mathsf{s}}[i]; \mathsf{q}[i]). \tag{3.2}$$

Intuitively, the distortion side information $\mathsf{q}[i]$ represents information such as the SNR of the $i$th source sample, the perceptual distortion masking, or other effects which make some components of a signal more or less important. Some possibilities for how the distortion side information might weight or otherwise modify the distortion measure on a sample-by-sample basis include single-letter distortion measures such as $d(\mathsf{s}[i], \hat{\mathsf{s}}[i]; \mathsf{q}[i]) = \mathsf{q}[i] \cdot (\mathsf{s}[i] - \hat{\mathsf{s}}[i])^2$ or $d(\mathsf{s}[i], \hat{\mathsf{s}}[i]; \mathsf{q}[i]) = |\mathsf{s}[i] - \hat{\mathsf{s}}[i]|^{\mathsf{q}[i]}$.

By allowing this richer class of distortion measures, we can more accurately represent the goals of a lossy compression system. Presumably, such accuracy should lead to better performance. For example, in a fully informed system where $\mathsf{q}\left[{}^{n}_{1}\right]$ is known to both encoder and decoder, this distortion side information could be used to allocate more bits to sensitive samples indicated by $\mathsf{q}[i]$. This intelligent bit allocation should outperform an uniformed system where all source samples are treated the same.

But what happens for a partially informed system where only the encoder knows $\mathsf{q}\left[{}^{n}_{1}\right]$? For example, in a sensor scenario it might be reasonable to model the encoder as having access to the SNR for each observation, but the receiver or decoder might not have such knowledge. Therefore, at first it might seem like the added benefit for a sophisticated distortion model would be outweighed by the need for the encoder to convey $\boldsymbol{q}\left[\begin{smallmatrix}n\\1\end{smallmatrix}\right]$ to the decoder. In Chapter 2, we show that this is often not the case. Specifically, we show that in many cases of interest it is possible for the partially informed system where only the encoder knows $\boldsymbol{q}\left[\begin{smallmatrix}n\\1\end{smallmatrix}\right]$ to obtain the same performance as the fully informed system where both encoder and decoder know $\boldsymbol{q}\left[\begin{smallmatrix}n\\1\end{smallmatrix}\right]$. Thus, in many cases we can gain the full benefit of sophisticated distortion side information models without the need to explicitly transmit $\boldsymbol{q}\left[\begin{smallmatrix}n\\1\end{smallmatrix}\right]$ from the encoder to the decoder.

While the results in Chapter 2 use random coding arguments, in this chapter we propose a constructive, practically implementable method to use distortion side information. Specifically, we describe lattice based lossy compression systems to exploit distortion side information known only at the encoder. We show that the proposed systems gain the full benefit of the distortion side information while requiring complexity which grows only linearly with the block length, $n$.

We focus on lattice based systems followed by entropy coding for a number of reasons. From a practical perspective, lattices have many advantages for source coding (see, *e.g.*, [78, 58, 117, 112, 41, 40, 130, 86, 197]). Indeed, an integer lattice followed by entropy coding is one of the most commonly used lossy compression systems. Furthermore, many results exists on constructing, encoding/decoding, analyzing, and using lattices such as [198, 64, 63, 165, 153, 2, 200, 168, 111, 177, 131]. Finally, lattices are a useful tool in analyzing the different types of gains possible in lossy compression. For example, [112] identifies the space-filling gain, shape gain, and memory gain as separate possible gains to pursue. Similarly, for high-rate quantizers, [58] identifies the granular gain and boundary as gains which can be achieved separately via lattices.

Essentially, we continue along these lines by introducing a type of transform across nested lattices and showing that this transform obtains the full benefit of the distortion side information. Thus the distortion side information gain is separate from both the boundary gain achieved by entropy coding and the granular gain achieved by selecting an appropriate sequence of nested lattices.

The rest of this chapter is organized as follows. We begin with some simple examples in Section 3.2 which convey the key insights of this chapter. These examples show how to design partially informed systems where only the encoder knows the distortion side information. Furthermore, these examples demonstrate that partially informed systems can achieve the same performance as fully informed systems where both encoder and decoder know the distortion side information. Next, we briefly establish some notation in Section 3.3. Section 3.4 presents our constructions for how lattice based quantizers which efficiently exploit distortion side information. While Section 3.4 focuses on the quantizer structure and codebook format, Section 3.5 we describes a sub-optimal but provably efficient encoding algorithm. Specifically, in Section 3.5.1 we describe the rule for mapping source and side information into the format specified in Section 3.4. Then in Section 3.5.3, we show that the proposed rule can be implemented with linear complexity. After introducing a statistical model for the source and side information in Section 3.5.4, we derive the rate-distortion performance for our proposed encoding rule in Section 3.5.6 and show that it matches the rate-distortion performance of the fully informed system in Section 3.5.5. We close with some concluding remarks in Section 3.6.

## 3.2 Two Simple Examples

We illustrate some different ways distortion side information can be efficiently used even when known only at the encoder.

### 3.2.1 Discrete Uniform Source

Consider the case where the source, $\mathsf{s}[i]$, corresponds to $n$ samples each uniformly and independently drawn from the finite alphabet $\mathcal{S}$ with cardinality $|\mathcal{S}| \geq n$. Let $\mathsf{q}[i]$ correspond to $n$ binary variables indicating which source samples are relevant. Specifically, let the distortion measure be of the form $d(s, \hat{s}; q) = 0$ if and only if either $q = 0$ or $s = \hat{s}$. Finally, let the sequence $\mathsf{q}[i]$ be statistically independent of the source with $\mathsf{q}[i]$ drawn uniformly from the $n$ choose $k$ subsets with exactly $k$ ones.

If the side information were unavailable or ignored, then losslessly communicating the source would require exactly $n \cdot \log |\mathcal{S}|$ bits. A better (though still sub-optimal) approach when encoder-only side information is available would be for the encoder to first tell the decoder which samples are relevant and then send only those samples. This would require $n \cdot H_b(k/n) + k \cdot \log |\mathcal{S}|$ bits where $H_b(\cdot)$ denotes the binary entropy function. Note that when the side information is known at both encoder and the decoder, then the overhead required in telling the decoder which samples are relevant could be avoided and the total rate required is exactly $k \cdot \log |\mathcal{S}|$. We will show that this overhead can in fact be avoided even without decoder side information.

Pretend that the source samples $\mathsf{s}[0]$, $\mathsf{s}[1]$, $\ldots$, $\mathsf{s}[n-1]$, are a codeword of an $(n, k)$ Reed-Solomon (RS) code (or more generally any MDS[1] code) with $\mathsf{q}[i] = 0$ indicating an erasure at sample $i$. Use the RS *decoding* algorithm to "correct" the erasures and determine the $k$ corresponding information symbols which are sent to the receiver. To reconstruct the signal, the receiver *encodes* the $k$ information symbols using the encoder for the $(n, k)$ RS code to produce the reconstruction $\hat{\mathsf{s}}[0]$, $\hat{\mathsf{s}}[1]$, $\ldots$, $\hat{\mathsf{s}}[n-1]$. Only symbols with $\mathsf{q}[i] = 0$ could have changed, hence $\hat{\mathsf{s}}[i] = \mathsf{s}[i]$ whenever $\mathsf{q}[i] = 1$ and the relevant samples are losslessly communicated using only $k \cdot \log |\mathcal{S}|$ bits.

As illustrated in Fig. 3-1, RS decoding can be viewed as curve-fitting and RS encoding can be viewed as interpolation. Hence this source coding approach can be viewed as fitting a curve of degree $k - 1$ to the points of $\mathsf{s}[i]$ where $\mathsf{q}[i] = 1$. The resulting curve can be specified using just $k$ elements. It perfectly reproduces $\mathsf{s}[i]$ where $\mathsf{q}[i] = 1$ and interpolates the remaining points.

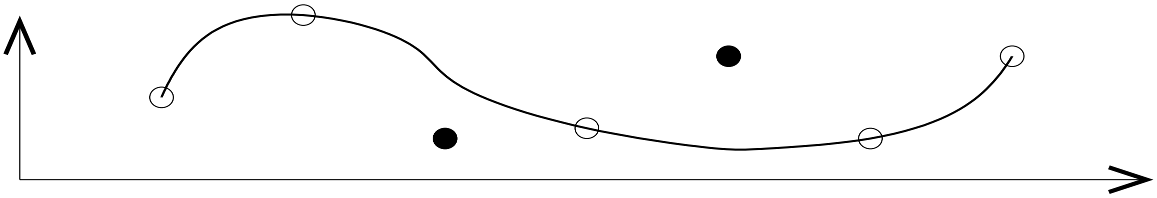


**Figure 3-1.** Losslessly encoding a source with $n = 7$ points where only $k = 5$ points are relevant (*i.e.*, the unshaded ones), can be done by fitting a fourth degree curve to the relevant points. The resulting curve will require $k$ elements (yielding a compression ratio of $k/n$) and will exactly reproduce the desired points.

[1] The desired MDS code always exists since we assumed $|\mathcal{S}| \geq n$. For $|\mathcal{S}| < n$, near MDS codes exist which give asymptotically similar performance with an overhead that goes to zero as $n \to \infty$.

### ■ 3.2.2 Weighted Quadratic Distortion

The previous example in Section 3.2.1 can be generalized to Gaussian sources with quadratic distortion. Specifically, provided that $d(\cdot, \cdot; q) = 0$ whenever $q = 0$ and is otherwise independent of $q$, then the technique in Section 3.2.1 can be generalized by replacing MDS codes with Fourier Transforms [124]. This requires a very special form for the distortion measure and may be of limited interest. In the rest of this section, we consider another example which illustrates how distortion side information may be used for a much wider range of distortion measures.

Consider quantizing a pair of source samples $(\mathsf{s}[1], \mathsf{s}[2])$ subject to the distortion constraint $d(s, \hat{s}; q) = q \cdot (s - \hat{s})^2$. Furthermore, imagine that the distortion side information is such that one source sample is sixteen times more important than the other. Specifically, let $(\mathsf{q}[1], \mathsf{q}[2]) = (1, 16)$ or $(\mathsf{q}[1], \mathsf{q}[2]) = (16, 1)$. In the following we consider how to design high-resolution quantizers when $\mathsf{q}$ is known at both encoder and decoder as well as when $\mathsf{q}$ is known only at the encoder.

#### Encoder & Decoder Know $\mathsf{q}$

For the moment, let us restrict attention to simple uniform scalar quantizers followed by entropy coding. If both encoder and decoder know the distortion side information, then the optimal strategy is straightforward: use a quantizer step-size of $\Delta$ for the more important source sample and a step-size of $4\Delta$ for the less important sample.

Fig. 3-2 illustrates this idea for quantizing two source samples, $\mathsf{s}[1]$ and $\mathsf{s}[2]$. If the first source sample is more important (*i.e.*, $\mathsf{q}[1] = 16$ and $\mathsf{q}[2] = 1$), then a denser quantizer is used for the first source sample and a coarser quantizer is used for the second source sample. If the second source sample is more important, the situation is reversed.

It is well-known that in the limit of small distortions, the average squared error for a step-size $\Delta$ uniform scalar quantizer is simply the variance of random variable uniformly distributed over the interval $(0, \Delta)$. Thus the expected distortion is approximately $16 \cdot \Delta^2/12$ for the more important sample and $(4\Delta)^2/12 = 16 \cdot \Delta^2/12$ (*i.e.*, the same) for the less important sample.

Provided that the source has a probability density function, it is straightforward to show that total number of bits required by the two quantizers is approximately

$$h(\mathsf{s}) - \log \Delta + h(\mathsf{s}) - \log 4\Delta \tag{3.3}$$

for a total average bit rate of

$$h(\mathsf{s}) - \log 2\Delta \tag{3.4}$$

bits per sample.

Essentially, using a fine-grained quantizer for the more important sample and a coarse-grained quantizer for the less important sample is like using a variable codebook and partition as illustrated in Fig. 3-3. Specifically, the codebook and partition in Fig. 3-3(a) correspond to the quantizer in Fig. 3-2(a) while the codebook and partition in Fig. 3-3(b) correspond to Fig. 3-2(b). In each case, the codebook and partition are designed to minimize the error in quantizing the more important sample. Note that, Fig. 3-3 illustrates that the decoder must know the distortion side information to know which point set to use in mapping a quantization index to a reconstruction of the source.

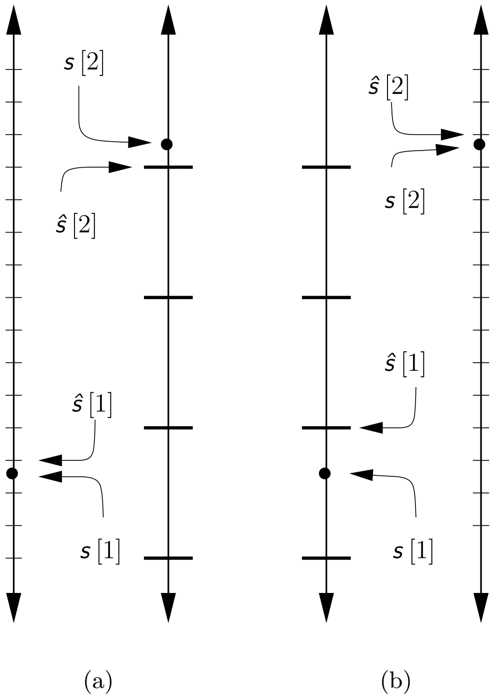


**Figure 3-2.** Two possible quantizers to use if both the encoder and decoder know the distortion side information. When the first source sample, $s\,[1]$, is more important as in (a), then a finer quantizer is used for it and a coarser quantizer is used for $s\,[2]$. In (b), the second source sample, $s\,[2]$ is more important and thus the situation is reversed.

### Only Encoder Knows $q$

How can we achieve similar performance when only the encoder knows the distortion side information? Ideally we wish to spend two extra bits on each sample where $q\,[i] = 16$ since these samples require more accurate description than samples with $q\,[i] = 1$. A naive approach in achieving this goal would be for the encoder to first use 1 bit per sample to tell the decoder whether or not a sample is important. Once this is accomplished then the structures in Figures 3-2 and 3-3 can be used. Of course, explicitly sending an extra bit to indicate whether a sample is important or not incurs an overhead which is not required when both encoder and decoder know $q$. This overhead makes the naive approach undesirable. Hence, we describe a system which can allocate two more bits of description to each important sample without requiring any overhead.

Our approach is to use a pair of shifted quantizers. We again use an example with two source samples. The encoder quantizes both source samples with a pair of quantizers of step-size $4\Delta$. The corresponding reconstructions are shifted by an offset of $\lambda \in \{0, \Delta, 2\Delta, 3\Delta\}$. The same offset is used for both quantizers. Thus the total number of bits required are the same as the fully informed system. Specifically, approximately $h(s) - \log 4\Delta$ bits are required to quantize each source sample to a grid of step-size $4\Delta$ plus an additional two bits to specify the shift $\lambda \in \{0, \Delta, 2\Delta, 3\Delta\}$. Thus the total average bit rate is the same as (3.4). The shift is chosen to minimize the error for the more important sample (*i.e.*, the one for which $q\,[i] = 16$). Using shifted quantizers in this manner has the effect of devoting

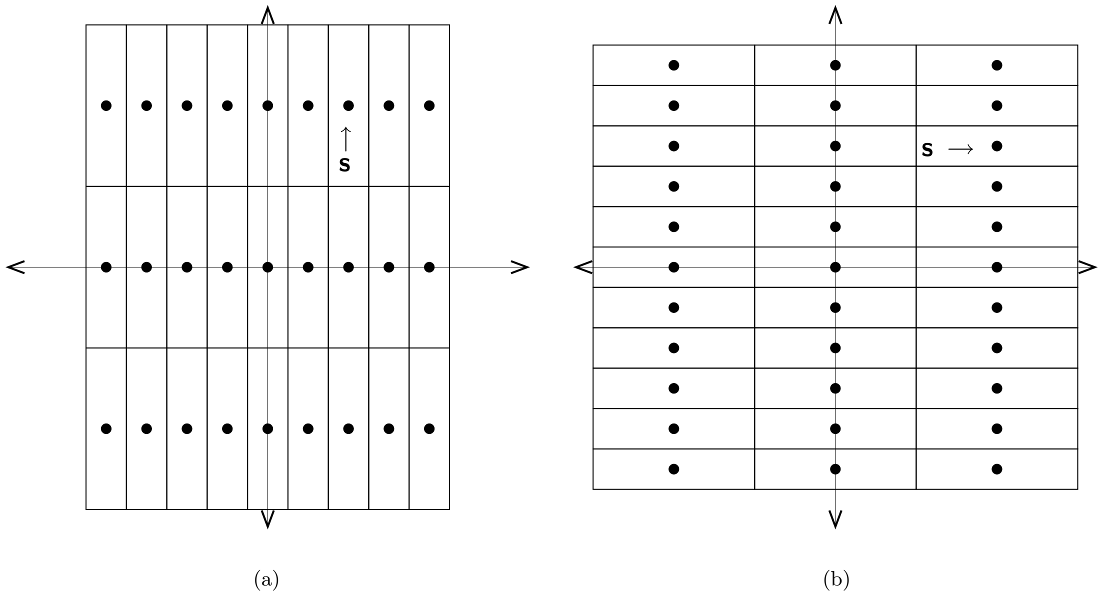


**Figure 3-3.** Variable codebook and partition view of the quantizers in Fig. 3-2 obtained by grouping the two samples into a vector $\mathbf{s} = (s[1], s[2])$. If the distortion side information indicates $s[1]$ is more important then the codebook and partition in (a) are used, while if the second sample is more important then (b) is used.

the extra two bits of information to describing the important sample without the need for the overhead required to describe which sample is important.

Fig. 3-4 illustrates this idea. In Fig. 3-4(a), a quantizer of step-size $4\Delta$ is used for both samples. By using an additional 2 bits to specify a shift of $\lambda = 3\Delta$ for the first sample, however, we are effectively able to use 2 more bits for the important sample and achieve both the same average rate and distortion as the fully informed system in Fig. 3-2(a). Alternatively, if the second source sample is important as in Fig. 3-4(b), the shift is chosen to minimize the error in quantizing $s[2]$ and we are again able to use 2 more bits for the important sample to achieve the same average rate and distortion as the fully informed system in Fig. 3-2(b).

Essentially, the idea of using a pair of quantizers with a single shift applied to both is like using a single, fixed set of codewords with a variable partition as illustrated in Fig. 3-5. Specifically, the partition in Fig. 3-5(a) corresponds to the quantizer in Fig. 3-4(a) while the partition in Fig. 3-5(b) corresponds to the quantizer in Fig. 3-4(b). In each case, the location of the points (*i.e.*, the codebook) is fixed while it is only the partition that changes. The partition depends on the distortion side information in the sense that the encoder finds the "nearest" quantization point where "nearest" depends on the distortion side information. Once the encoder has chosen the best quantization point and described it to the decoder, the decoder can produce a quantized reconstruction of the source without knowing the distortion side information.

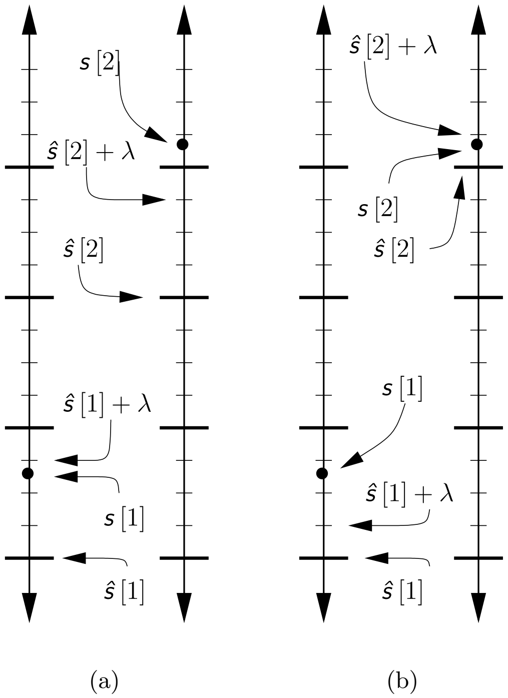


**Figure 3-4.** Two possible quantizers to use if only the encoder knows the distortion side information. The structure in (a) is used if the first sample is more important and (b) is used if the second sample is more important. In either case, each source sample is quantized to a grid of size $4\Delta$ represented by the wide tick marks. Next, both reconstructions are shifted by an offset, $\lambda \in \{0, \Delta, 2\Delta, 3\Delta\}$, where $\Delta$ corresponds to the fine spacing between narrow tick marks. When the first source sample, $s[1]$, is more important as in (a), $s[1]$ is quantized to the bottom wide tick mark and an offset of $\lambda = 3\Delta$ is selected. The error of the resulting reconstruction, $\hat{s}[1] + \lambda$, essentially corresponds to the fine grid size, $\Delta$. Conversely, the less important sample is quantized to the second wide tick mark from the top so that the resulting error is proportional to the coarse grid size, $4\Delta$. The situation is reversed in (b) where the second source sample is more important and thus a shift of $\lambda = \Delta$ is selected to minimize the error in quantizing $s[2]$.

## ■ 3.3 Notation

Vectors are denoted in bold (*e.g.*, $\mathbf{x}$). The $i$th element of a sequence is denoted as $x[i]$ if it is scalar or $\mathbf{x}[i]$ if it is a vector. Similarly, a subsequence of elements ranging from $i$ to $j$ is denoted as $\mathbf{x}\left[{j \atop i}\right]$ for sequences of vectors or $x\left[{j \atop i}\right]$ for sequences of scalars. Random variables are denoted using the sans serif font (*e.g.*, $\mathsf{x}$) while random vectors are denoted with bold sans serif (*e.g.*, $\boldsymbol{\mathsf{x}}$). We denote mutual information, entropy, and expectation as $I(\mathsf{x};\mathsf{y})$, $H(\mathsf{x})$, $E[\mathsf{x}]$. Calligraphic letters denote sets (*e.g.*, $s \in \mathcal{S}$) with $|\cdot|$ denoting the cardinality of its argument. Finally, for any length $n$ sequence $q\left[{n \atop 1}\right]$, we denote the number of occurrences of the symbol $i$ as $\#[q\left[{n \atop 1}\right] \sim i]$. In the Information Theory literature, $\#[q\left[{n \atop 1}\right] \sim i]$ is sometimes refereed to as the type of $q\left[{n \atop 1}\right]$.

## ■ 3.4 Lattice Quantizer Structures

In this section we describe how to generalize the construction in Section 3.2 to a wide variety of source and distortion measures. Specifically, our main goal is to construct efficient

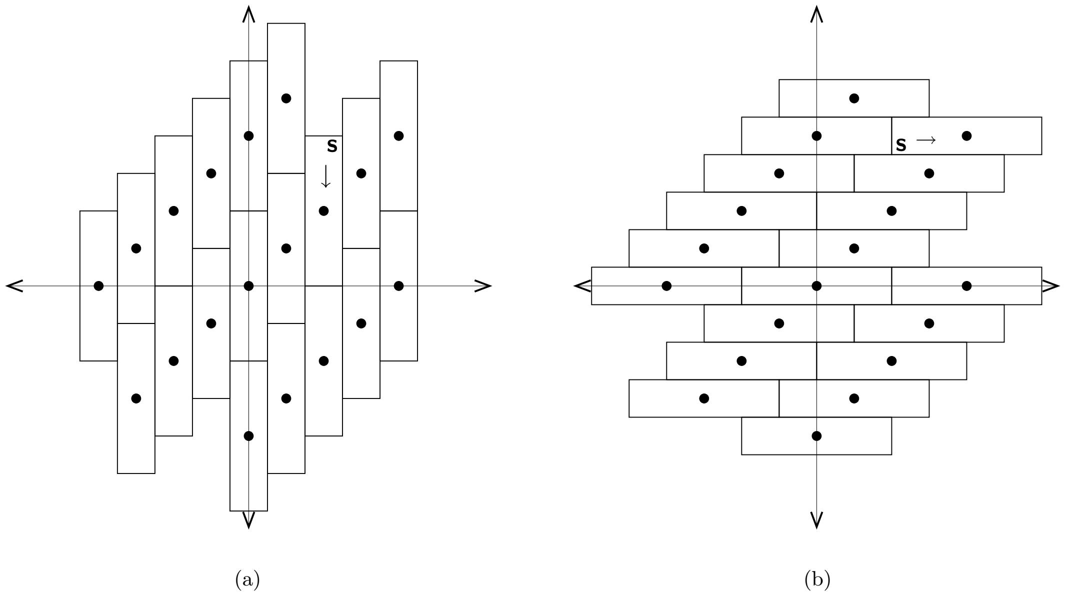


**Figure 3-5.** A *fixed* codebook/*variable* partition view of the quantizers in Fig. 3-4 obtained by grouping the two source samples into a vector $\mathsf{s} = (s\,[1]\,, s\,[2])$. If the distortion side information indicates $s\,[1]$ is more important then the partition in (a) is used, while if the second sample is more important then partition (b) is used. In either case, the source is mapped to the codeword which minimizes the average error weighted by the distortion side information. Note that while in Fig. 3-3 both the partition and codebook change as a function of the distortion side information, here only the partition depends on the side information and the location of the points (*i.e.*, the codebook) is fixed.

quantizers when the distortion side information is known only at the encoder. In measuring efficiency, we will compare a partially informed system where only the encoder knows the distortion side information to a fully informed system where both encoder and decoder know the distortion side information. An efficient partially informed system is one which either exactly or asymptotically achieves the same performance as the corresponding fully informed system.

We consider a quantizer or lossy compression system to consist of an encoder and decoder which work as follows. The encoder is presented with a length $n$ source sequence $\mathbf{s}\left[{}^{n}_{1}\right]$ and a length $n$ distortion side information sequence $q\left[{}^{n}_{1}\right]$. The encoder produces an index which is stored or transmitted to a decoder. In a fully informed system, the decoder receives both the encoder output as well as $q\left[{}^{n}_{1}\right]$. In a partially informed system the decoder receives the index but does not know $q\left[{}^{n}_{1}\right]$. In either case, the error for the reconstruction $\hat{\mathbf{s}}\,{}^{n}_{1}$ is computed by averaging the single-letter distortion measure, $d(\cdot,\cdot;\cdot)$, to obtain

$$d(\mathbf{s}\left[{}^{n}_{1}\right], \hat{\mathbf{s}}\left[{}^{n}_{1}\right]; q\left[{}^{n}_{1}\right]) = \frac{1}{n}\sum_{i=1}^{n} d(\mathbf{s}\,[i]\,, \hat{\mathbf{s}}\,[i]\,; q\,[i]). \tag{3.5}$$

The key insight in developing the example in Section 3.2.2 is the implicit use of a coset decomposition. Specifically, we used the fact that the integer lattice, $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, can be decomposed into four shifts of the sub-lattice of integers which are multiples of four, $4\mathbb{Z} = \{\ldots, -8, -4, 0, 4, 8, \ldots\}$. Both the fully informed system in Section 3.2.2 and the partially informed system in Section 3.2.2 were constructed by essentially using the $\mathbb{Z}$ lattice to quantize the more important sample and using the $4\mathbb{Z}$ lattice to quantize the less important sample. Hence, to develop quantizers which can efficiently use encoder side information for a variety of sources and distortion measures, we develop quantizers based on lattices. We begin by briefly reviewing some lattice terminology. Further details on lattices can be found in [63, 41].

### ■ 3.4.1 A Brief Review of Lattices and Cosets

A real, $r$-dimensional lattice, $\Lambda \subset \mathbb{R}^r$, is a regular array of points in space which forms a group under vector addition.[2] The nesting idea used in Section 3.2.2 can be captured by sub-lattices. Specifically, a sub-lattice $\Lambda' \subset \Lambda$ is any subgroup of $\Lambda$ with respect to vector addition. We refer to a sequence of sub-lattices $\Lambda(1) \subset \Lambda(2) \subset \ldots \subset \Lambda(L)$ of the base lattice $\Lambda(L)$ as a nested lattice. Throughout the chapter, the reader may find it useful to keep in mind integer nested lattices as a concrete embodiment of a nested lattice. Specifically, let $\Lambda(j)$ be integer multiples of $2^{L-j}$. Thus $\Lambda(L)$ is simply the integers $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, $\Lambda(L-1)$ consists of even integers $2\mathbb{Z} = \{\ldots, -4, -2, 0, 2, 4, \ldots\}$, $\Lambda(L-2)$ consists of multiples of four, $4\mathbb{Z}$, and so on.

For a pair of lattices $\Lambda' \subset \Lambda$, we denote the partition of $\Lambda$ into equivalence classes modulo $\Lambda'$ as $\Lambda/\Lambda'$. By taking one element from each equivalence class we obtain a system of coset representatives denoted by $[\Lambda/\Lambda']$. By using a coset decomposition for the partition we can uniquely specify a point $\lambda$ in the base lattice $\Lambda$ by the sum $\lambda' + c$ where $\lambda'$ is an element of the sub-lattice $\Lambda'$ and $c$ is the coset representative in $[\Lambda/\Lambda']$. For the example where $\Lambda(L) = \mathbb{Z}$ and $\Lambda(L-1) = 2\mathbb{Z}$, we can choose $[\Lambda(L)/\Lambda(L-1)] = \{0, 1\}$. This allows us to represent any integer $z \in \mathbb{Z}$ as $z = e + c$ where $e$ is an even integer and $c \in \{0, 1\}$.

We can continue this idea further. Instead of the representation

$$\Lambda(L) = \Lambda(L-1) + [\Lambda(L)/\Lambda(L-1)] \tag{3.6a}$$

where any element in $\Lambda(L)$ is described by an element of $\Lambda(L-1)$ and a coset representative in $[\Lambda(L)/\Lambda(L-1)]$, we can use the further decomposition

$$\Lambda(L-1) = \Lambda(L-2) + [\Lambda(L-1)/\Lambda(L-2)]. \tag{3.6b}$$

Combining (3.6a) and (3.6b) yields the representation

$$\Lambda(L) = \Lambda(L-2) + [\Lambda(L-1)/\Lambda(L-2)] + [\Lambda(L)/\Lambda(L-1)]. \tag{3.6c}$$

For example, setting $\Lambda(L) = \Delta\mathbb{Z}$, with $\Lambda(L-1) = 2\Delta\mathbb{Z}$ and $\Lambda(L-2) = 4\Delta\mathbb{Z}$ yields the decomposition used in Section 3.2.2. Specifically, to describe an element of a grid of step-size $\Delta$, we can instead describe an element of a grid of step-size $4\Delta$ plus an element from $\{0, \Delta\}$ plus an element from $\{0, 2\Delta\}$. Taken to its conclusion this notation allows us to use the partition chain

$$\Lambda(L)/\Lambda(L-1)/\ldots/\Lambda(1) \tag{3.7}$$

to represent any element of $\Lambda(L)$ with an element of $\Lambda(1)$ combined with the $L-1$ coset representatives from each partition.

[2] For simplicity we focus on lattices, but most of our constructions and analysis only require the group property and some notion of uniformity such as [65]. Thus, our ideas can be easily extended to other spaces such as quantizing phases or angles.

## 3.4.2 The Fully Informed System

First we consider a fully informed system for quantizing the source sequence $\mathbf{s}\left[{}^{n}_{1}\right]$ where each source sample, $\mathbf{s}[i]$, is an $r$-dimensional vector. The fully informed system is designed to treat $L$ different kinds of samples (*e.g.*, if $q$ takes exactly $L$ different values) and consists of a sequence of $r$-dimensional nested lattices

$$\Lambda(1) \subset \Lambda(2) \subset \Lambda(3) \subset \ldots \subset \Lambda(L). \tag{3.8}$$

Intuitively, the finest lattice $\Lambda(L)$ is used to quantize the most important samples, while the coarser lattice, $\Lambda(L-1)$, is used for the next most important class of samples, and so on until $\Lambda(1)$ is used for the least important samples.

To quantize a source sequence $\mathbf{s}\left[{}^{n}_{1}\right]$, we first classify each source sample, $\mathbf{s}[i]$, into one of $L$ classes according to the value of $q[i]$. For example, if $q \in \{1, 2, \ldots, L\}$, we can simply represent the class for $\mathbf{s}[i]$ by the value of $q[i]$. To simplify the exposition we assume that this is the case. [3] Then $\mathbf{s}[i]$ is quantized to the nearest point in lattice $\Lambda(q[i])$, which we represent as $\lambda_{\Lambda(q[i])}(i)$. Finally, the resulting quantization indexes are entropy coded and described to the receiver. The system described in Section 3.2.2 is one example of a fully informed system where $L = 2$, $\Lambda(2) = \Delta\mathbb{Z}$, $\Lambda(1) = 4\Delta\mathbb{Z}$, and $d(s, \hat{s}; q) = q^4 \cdot (s - \hat{s})^2$.

## 3.4.3 The Partially Informed System

To facilitate comparison, we construct a partially informed system using the same lattice and lattice partition chain as for the fully informed system. Specifically, our basic tool is a sequence of $L$ $r$-dimensional nested lattices as in (3.8). Our goal is to match the performance of the corresponding fully informed system described in Section 3.4.2. Note that one simple and sub-optimal approach would be for the encoder to simply send $q\left[{}^{n}_{1}\right]$ to the decoder and then use the quantizer for the fully informed system. This would incur an overhead of $H(\boldsymbol{q})$ bits per sample. Instead, we describe a quantizer which uses the same lattices as the fully informed system in a somewhat different manner in order to avoid the overhead of $H(\boldsymbol{q})$ required in communicating $\boldsymbol{q}$.

### Codebook Structure and Decoding

To quantize the source sequence $\mathbf{s}\left[{}^{n}_{1}\right]$ with the distortion side information $q\left[{}^{n}_{1}\right]$, the encoder sends three different pieces of information:

1. A list of the number of samples in $q\left[{}^{n}_{1}\right]$ taking the value $i$ (denoted as $\#[q\left[{}^{n}_{1}\right] \sim j]$)
2. A sequence of $n$ elements from the coarsest lattice $\Lambda(1)$ (this sequence is denoted $\lambda_{\Lambda(1)}(i)$ with $i \in \{1, 2, \ldots, n\}$)
3. A table of coset representatives, $c[i, j]$, where $c[i, j]$ is the coset representative for the partition $[\Lambda(j+1)/\Lambda(j)]$ for sample $i$ with $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, L-1\}$

Postponing for the moment how each of these components is determined and efficiently encoded, once this encoding of the source $\mathbf{s}\left[{}^{n}_{1}\right]$ is received, the decoder reconstructs the

[3] Instead, if the number of possible values of $q[i]$ is large or if $q$ takes values in a continuum, then a more complicated classifier may be required. In such cases, we could transform the original problem into one with a new type of distortion side information, $q'$, and a new distortion measure where there are only $L$ possible values for $q'$. Hence our assumption incurs no essential loss of generality but simplifies notation.

sequence $\hat{\mathbf{s}}\left[{n \atop 1}\right]$ by setting

$$\hat{\mathbf{s}}[i] = \lambda_{\Lambda(1)}(i) + \sum_{j=1}^{L-1} c[i,j]. \tag{3.9}$$

### Encoding $\#[q\left[{n \atop 1}\right] \sim j]$

The number of occurrences of the sample $j$ in $q\left[{n \atop 1}\right]$ can be encoded by via entropy coding (*e.g.*, Huffman coding or Arithmetic coding) if a statistical characterization of $\boldsymbol{q}$ is available. Even without any statistical characterization of $\boldsymbol{q}$, however, $\#[q\left[{n \atop 1}\right] \sim j]$ can be encoded simply as a list of $L-1$ integers from 0 to $n$ where the $i$th value in the list is simply $\#[q\left[{n \atop 1}\right] \sim j]$.[4] Thus describing $\#[q\left[{n \atop 1}\right] \sim i]$ requires at most

$$(L-1) \cdot \lceil \log_2 n \rceil \tag{3.10}$$

bits.

### Encoding $\lambda_{\Lambda(1)}(i)$

The lattice elements in the base lattice can be encoded by entropy coding (*e.g.*, Huffman coding or Arithmetic coding).

### Encoding $c[i,j]$

The key step in our construction is efficiently encoding the table of coset representatives. Before describing our encoding method we first illustrate the intuition behind our construction.

For the least important samples where $q[i] = 1$, the fully informed system uses the coarsest lattice, $\Lambda(1)$. Thus, for these least important samples, the partially informed system should ideally not spend any more bits to describe the source than the fully informed system did. Therefore, the partially informed system should ideally spend zero bits to not describe $c[i,j]$ at all when $q[i] = 1$. Similarly, when $q[i] = 2$, the fully informed system uses the second coarsest lattice, $\Lambda(2)$ and so the partially informed system should ideally only describe $c[i,1]$ and not attempt to describe $c[i,j]$ for $j > 1$. This reasoning extends to the remaining coset representatives.

In order to achieve this goal we use erasure quantizing codes which are in some sense dual to erasure correction codes. Specifically, an $(n,k,e)$ erasure quantizing code over an alphabet $\mathcal{X}$ is a set of codewords $\mathbf{c} \in \mathcal{C}$ with the following properties:

1. Each codeword is a vector of $n$ elements from the alphabet $\mathcal{X}$.

2. The code has dimension $k$ in the sense that each element of $\mathcal{C}$ can be uniquely specified with $k$ letters from $\mathbf{c}$.

3. For any source vector of length $n$, if at least $e$ positions are marked as erasures, then there exists at least one codeword in $\mathcal{C}$ which matches the remaining $n-e$ positions exactly.

[4] The value of $\#[q\left[{n \atop 1}\right] \sim L]$ does not need to be explicitly encoded since it can be implicitly determined as $n - \sum_{j=1}^{L-1} \#[q\left[{n \atop 1}\right] \sim j]$.

For example, the familiar single parity check linear code where $\mathcal{C}$ consists of all vectors whose modulo-$q$ sum is zero is an $(n, n-1, 1)$ erasure quantizing code. In general, any maximum distance separable $(n, k)$ erasure correcting code is an $(n, k, n-k)$ error correction code and vice versa. Thus we will refer to $(n, k, n-k)$ erasure quantizing codes as maximum distance separable (MDS). For large enough $n$, it is possible to construct $(n, k, e)$ near-MDS erasure quantizing codes where $e/n$ converges to $1 - k/n$. Furthermore, constructions of near-MDS erasure quantizing codes exist whose encoding/decoding complexity is linear in the block length [125].

We use erasure quantizing codes to encode $c[i, j]$ as follows. For a given $j$, instead of describing $c[i, j]$ with $i$ ranging from 1 to $n$ directly, we use a near-MDS erasure quantizing code with length $n$ and dimension $k = \#[q\left[{}^{n}_{1}\right] \sim j]$ over an alphabet of size $|\Lambda(j+1)/\Lambda(j)|$. The $j$th erasure quantizing code is used to describe the coset representatives only for those samples where $q\,[i] > j$ and the coset representatives for samples with $q\,[i] \leq j$ are marked as erasures. The following example best illustrates how erasure quantizing codes are used to efficiently encode the coset representatives.

### A Brief Example

So far we have only described the encoding format and not specified how the encoder might map its inputs into this format. The mapping of inputs into this format will be discussed in Section 3.5. Thus, in this example, we focus on illustrating the intuition behind the codebook format described previously.

Let $L = 3$, with $\Lambda(1)$ being the integers which are multiples of four, $\Lambda(2)$ being the even integers, and $\Lambda(3)$ being simply the integers (*i.e.*, $\Lambda(1) = 4\mathbb{Z}$, $\Lambda(2) = 2\mathbb{Z}$, and $\Lambda(3) = \mathbb{Z}$). Furthermore, assume we are interested in a block of length $n = 3$ with

$$q\left[{}^{3}_{1}\right] = (2, 3, 1) \text{ and } s\left[{}^{3}_{1}\right] = (11.8, 1.1, 8.7). \tag{3.11}$$

First, $\#[q\left[{}^{3}_{1}\right] \sim i] = (1, 1, 1)$ is described to the decoder.

Then, each source sample is quantized to the base lattice, $\Lambda(1)$, yielding $\lambda_{\Lambda(1)}(i) = (8, 0, 4)$ and $\lambda_{\Lambda(1)}(i)$ is described to the decoder.[5]

The first and second sample are more important than the third sample since they have higher values of $q\,[i]$ and thus we would ideally like to describe these samples with an element from a finer lattice. We can refine the quantization values in the base lattice for these samples by specifying the coset representatives for the partition $[\Lambda(2)/\Lambda(1)]$ only for the first two samples. We accomplish this by using a $(3, 2, 1)$ binary erasure quantizing code (*e.g.*, a binary single parity check code). Specifically, we consider the coset representative for the third sample to be an erasure and find a codeword of the $(3, 2, 1)$ binary erasure quantizing code to match the coset representatives for the first two samples. Thus $c[i, 1] = (1, 0, 1)$.

Finally, the second source sample has the highest value of $q$ and thus we would ideally like to describe this sample with an element from the finest lattice. We can refine the quantization value for this sample in $\Lambda(2)$ by specifying a coset representative for the partition $[\Lambda(3)/\Lambda(2)]$ only for the second sample. We accomplish this by using a $(3, 1, 2)$ binary erasure quantizing code (*e.g.*, a binary repetition code). Thus $c[i, 2] = (1, 1, 1)$. The resulting reconstruction is

$$\hat{\mathbf{s}}\left[{}^{3}_{1}\right] = (8, 0, 4) + 2 \cdot (1, 0, 1) + (1, 1, 1) = (11, 1, 7) \tag{3.12}$$

[5] Note that $s\,[3]$ is quantized to 4 instead of the seemingly closer 8. This is to account for the shifts caused by later choices for coset representatives and discussed in more detail in Section 3.5.1.

with an absolute error of $(0.8, 0.1, 1.7)$.

If instead $q\left[{}^{3}_{1}\right]$ was $(2,1,3)$ we would choose the base quantization in $\Lambda(1)$ to be $\lambda_{\Lambda(1)}(i) = (8,0,8)$ with $c[i,1] = 2 \cdot (1,1,0)$ and $c[i,2] = (1,1,1)$ to get $\hat{\mathbf{s}}\left[{}^{3}_{1}\right] = (11,3,9)$ for an absolute error of $(0.8, 1.9, 0.7)$. In both cases, we see that the use of erasure quantizing codes allow us to more accurately describe samples with higher values of $q\left[{}^{3}_{1}\right]$.

## ■ 3.5 Efficient Encoding and Performance

In Section 3.4.3 we essentially described the decoder or codebook for a partially informed system. In particular, we did not specify how the source and distortion side information are mapped into the required quantized bit stream. Obviously, this mapping should be performed in a manner which minimizes the resulting distortion between the source and the quantized reconstruction. Since the partially informed system is a composite lattice, mapping the source and distortion side information into the quantized bit stream to minimize the distortion essentially corresponds to finding a point in the composite lattice which is nearest to the source.

In general finding the closest point in a lattice is an NP-hard problem [54]. Various heuristics exist which sometimes work well [54,82,2,178], but even these heuristics generally have super-linear (usually cubic) complexity. Hence, we describe a sub-optimal encoding procedure which requires only linear complexity in the block length but nonetheless produces a distortion which is asymptotically the same as the distortion for the fully informed system.

### ■ 3.5.1 The Encoding Algorithm

Our encoding algorithm is essentially a type of multistage quantization. First, we quantize the most important samples with the $q[i] = L$ where $L$ is the largest possible value for the distortion side information. This process requires quantizing the corresponding $\mathbf{s}[i]$ to the finest lattice $\Lambda(L)$. We then describe the resulting points in the finest lattice with a coset representation $c[i,j]$. But in order to efficiently encode $c[i, L-1]$ and avoid spending bits on describing $c[i, L-1]$ for values of $i$ where $q[i] < L$, we use erasure quantizing codes. Finally, we subtract the resulting coset representatives from the source so that later stages work with the remaining quantization error. This completes the encoding for $c[i, L-1]$. We then repeat the process for samples with $q[i] = L-1$ and so on. One key feature of this process is that in the $v$th stage, it allows us to only consider samples with $q[i] = L+1-v$ and decouple the various lattices. A detailed description of the encoding algorithm follows:

1. Initialize the coset representatives table, $c[i,j]$, to "erasure" for all $i \in \{1, 2, \dots, n\}$ and $j \in \{1, 2, \dots, L-1\}$.

2. Initialize $v$ to $v = L$.

3. Quantize all source samples with $q[i] = v$ to $\Lambda(v)$, *i.e.*, compute $\lambda_{\Lambda(v)}(i)$ for all $i$ with $q[i] = v$.

4. Separate each of the values for $\lambda_{\Lambda(v)}(i)$ found in the previous step into a coset decomposition of the form

$$\lambda_{\Lambda(v)}(i) = \lambda_{\Lambda(1)}(i) + \sum_{j=1}^{v-1} c[i,j] \in [\Lambda(j+1)/\Lambda(j)]. \tag{3.13}$$

5. Find a codeword in the $(n, \#[q\left[\begin{smallmatrix} n \\ 1 \end{smallmatrix}\right] \sim v])$ near-MDS erasure quantizing code over an alphabet of size $|\Lambda(v)/\Lambda(v-1)|$ which exactly matches all unerased values in $c[i, v-1]$ for $i$ ranging from 1 to $n$.

6. Subtract the resulting coset representatives from the source, *i.e.*, for each $i$ set $\mathbf{s}\,[i]$ to be $\mathbf{s}\,[i] - c[i, v-1]$.

7. Decrease $v$ by 1.

8. If $v \geq 1$ return to step 3.

9. Entropy code the description of $\lambda_{\Lambda(1)}(i)$ for $i$ ranging from 1 to $n$ (*e.g.*, using Huffman coding, Lempel-Ziv coding, arithmetic coding, *etc.*).

10. The algorithm is complete and $\mathbf{s}\,[i]$ now contains the difference between the original source and the quantized value computed by the algorithm.

### ■ 3.5.2 An Example of Encoding

To clarify the encoding algorithm, we again return to the first example in Section 3.4.3 with $q\left[\begin{smallmatrix} 3 \\ 1 \end{smallmatrix}\right]$ and $s\left[\begin{smallmatrix} 3 \\ 1 \end{smallmatrix}\right]$ as in (3.11).

In the first pass of the algorithm, we set $v = 3$ and quantize $s\,[2] = 1.1$ to the nearest point in the $\mathbb{Z}$ lattice to obtain $\lambda_{\Lambda(3)}(2) = 1$. The resulting coset representation has $c[2, 2] = 1$ which must be encoded by a (3,1,2) erasure quantizing code (*i.e.*, a binary repetition code) to yield $c[\begin{smallmatrix} 3 \\ 1 \end{smallmatrix}, 2] = 1$. Thus, in step 6 of the encoding algorithm, we subtract these coset representatives from the source to obtain

$$s\left[\begin{smallmatrix} 3 \\ 1 \end{smallmatrix}\right]' = (10.8, 0.1, 7.7) \tag{3.14}$$

after the first stage completes.

In the second pass, we set $v = 2$ and quantize $s\,[1]'$ to the nearest point in the $2\mathbb{Z}$ lattice to obtain $\lambda_{\Lambda(2)}(1) = 10$. Note that, in the previous pass we already determined that $\lambda_{\Lambda(3)}(2) = 1$ which in the coset decomposition

$$\lambda_{\Lambda(3)}(2) = \lambda_{\Lambda(2)}(2) + c[2, 2] \tag{3.15}$$

implies that $\lambda_{\Lambda(2)}(2) = 0$. The resulting coset representatives for $\lambda_{\Lambda(2)}(1)$ and $\lambda_{\Lambda(2)}(2)$ are $c[1, 1] = 2$ and $c[2, 1] = 0$. These coset representatives must be encoded by a (3,2,1) erasure quantizing code (*i.e.*, a binary single parity check code) to yield $2 \cdot (1, 0, 1)$. Subtracting this from the source yields

$$s\left[\begin{smallmatrix} 3 \\ 1 \end{smallmatrix}\right]'' = (8.8, 0.1, 5.7) \tag{3.16}$$

at the conclusion of the second stage.

In the third pass, we set $v = 1$ and quantize $s\,[3]''$ to the nearest lattice point in the $4\mathbb{Z}$ lattice to obtain $\lambda_{\Lambda(1)}(3) = 4$. Note that in the previous two passes we already quantized the other source samples which, through the appropriate coset decomposition, yields $\lambda_{\Lambda(1)}(1) = 8$ and $\lambda_{\Lambda(1)}(2) = 0$.

Thus after the encoding process terminates, we obtain the reconstruction in (3.12).

### 3.5.3 Complexity Analysis

**Theorem 11.** *Let $M(n)$ be the complexity of finding a codeword in step 5 of the algorithm in Section 3.5.1 and let $E(n)$ be the complexity of entropy coding $\lambda_{\Lambda(1)}(i)$ in step 9. Then the complexity of this algorithm is*

$$\mathcal{O}\left(E(n) + L \cdot (M(n) + n)\right). \tag{3.17}$$

*Proof.* The operations in steps 3, 4, and 6 each require $\mathcal{O}(n)$ complexity since they perform a single operation for each source sample. The loop is performed exactly $L$ times and so combining this with the assumed complexity for steps 5 and 9 yields (3.17). □

**Corollary 2.** *If linear-time erasure quantizing codes are used (*e.g.*, the ones constructed in [125] along with a linear time entropy coding algorithm, then the complexity of the algorithm in Section 3.5 is linear in the block length $n$.*

### 3.5.4 Statistical Models and Distortion Measures

In order to the expected bit rate required using the proposed structures and the resulting average distortion we need to introduce a statistical model for the source and distortion side information. For simplicity, we consider a memoryless source with memoryless distortion side information which is independent of the source.[6] Specifically, we model the source $\mathbf{s}\left[{n \atop 1}\right] = (\mathbf{s}[1], \mathbf{s}[2], \ldots, \mathbf{s}[n])$ as a sequence of either vectors or scalars taking values in the alphabet $\mathcal{S}$. The source is generated according to the independent and identically distributed (i.i.d.) probability law

$$p_{\mathbf{s}\left[{n \atop 1}\right]}(\mathbf{s}\left[{n \atop 1}\right]) = \prod_{i=1}^{n} p_{\mathbf{s}}(\mathbf{s}[i]). \tag{3.18}$$

Similarly, we model the distortion side information $q\left[{n \atop 1}\right] = (q[1], q[2], \ldots, q[n])$ as a sequence of scalars taking values in the finite alphabet $\mathcal{Q}$. The distortion side information is independent of the source and generated according to the i.i.d. probability law

$$p_{q\left[{n \atop 1}\right]|\mathbf{s}\left[{n \atop 1}\right]}(q\left[{n \atop 1}\right] | \mathbf{s}\left[{n \atop 1}\right]) = \prod_{i=1}^{n} p_{q}(q[i]). \tag{3.19}$$

Furthermore, we will focus on the special case of difference distortion measures where, for fixed $q$, the distortion between a source, $\mathbf{s}$, and a reconstruction, $\hat{\mathbf{s}}$, only depends on the difference $\mathbf{s} - \hat{\mathbf{s}}$:

$$d(\mathbf{s}, \hat{\mathbf{s}}; q) = \rho(\mathbf{s} - \hat{\mathbf{s}}; q) \tag{3.20}$$

#### Admissible Lattices, Sources, and Distortion Measures

Results in the theory of lattice quantization are often developed in terms of the Voronoi region of a basic cell of the lattice [40, 63, 112, 64, 197, 58]. Specifically, for any lattice point $\lambda \in \Lambda(q)$, and the associated distortion measure, $d(\mathbf{s}, \hat{\mathbf{s}}; q)$, the Voronoi region of a lattice

[6] Extending our results to sources with memory is straightforward and simply requires replacing algorithms/results for losslessly compressing memoryless sources with algorithms/results for losslessly compressing sources with memory. In contrast, exploiting memory in the distortion side information or correlation between the source and distortion side information to further reduce the expected bit rate seems to require additional tools.

point $\mathcal{V}\left[\lambda \in \Lambda(q), d(\mathbf{s}, \hat{\mathbf{s}}; q)\right]$ is the set of all points closest to that lattice point.[7] For a difference distortion measure, each lattice point has the same Voronoi region.

Let us define the random variable $\mathbf{u}_{\Lambda(q),\rho(\cdot,q)}$ as being uniformly distributed over this Voronoi region. Under certain mild technical conditions, the distribution of the difference between the source and its quantized reconstruction asymptotically approaches that of $\mathbf{u}_{\Lambda(q),\rho(\cdot,q)}$. Intuitively, as the density of the lattice increases, the source distribution becomes roughly constant and uniform over each Voronoi region allowing us to focus on $\mathbf{u}_{\Lambda(q),\rho(\cdot,q)}$. Alternatively, a dithered quantizer can be used where a uniform random variable is added to the source prior to quantization and subtracted out during reconstruction. The dither makes the quantization error uniformly distributed over the Voronoi region for all resolutions at the cost of a slight increase in rate [92, 199, 80].

The technical details of how the quantization error approaches a uniform distribution are somewhat involved (see [176, 134] for some details). Since our main goal is to present a quantizer structure and the associated algorithm we adopt a strategy similar to [58, 192, 74, 72] and essentially assume that quantization error is uniform and develop the consequences of this assumption. Specifically, we introduce the following definition:

**Definition 2.** *A sequence of lattices, $\Lambda(1)$, $\Lambda(2)$, …, $\Lambda(L)$, a source distribution $p_{\mathbf{s}}(\mathbf{s})$ and difference distortion a measure $d(\mathbf{s}, \hat{\mathbf{s}}; q) = \rho(\mathbf{s} - \hat{\mathbf{s}}; q)$ are admissible if*

1. *For each $q$ the Voronoi region of $\Lambda(q)$ with respect to $\rho(\cdot; q)$ is well defined and bounded.*

2. *The quantization error in mapping $\mathbf{s}$ to the nearest point $\lambda \in \Lambda(q)$ with respect to $\rho(\cdot; q)$ asymptotically approaches a uniform distribution over the Voronoi region independent of $\mathbf{s}$ and $\lambda$.*

## ■ 3.5.5 Rate-Distortion For Fully Informed Systems

**Theorem 12.** *For an admissible source, nested lattice, and distortion measure, the average rate and distortion for a fully informed system asymptotically approach the following parametric form in the high-resolution limit:*

$$D_{\mathrm{FI}} \triangleq E\left[\frac{1}{n}\sum_{i=1}^{n} d(\mathbf{s}[i], \hat{\mathbf{s}}[i]; \boldsymbol{q}[i])\right] \tag{3.21}$$

$$\rightarrow \sum_{j=1}^{L} p_{\boldsymbol{q}}(j) \cdot E[\rho(\mathbf{u}_{\Lambda(q),\rho(\cdot,q)}; \boldsymbol{q}) | \boldsymbol{q} = j] \tag{3.22}$$

$$R_{\mathrm{FI}} \triangleq \frac{1}{n} H(\hat{\mathbf{s}}\left[{}^{n}_{1}\right] | \boldsymbol{q}\left[{}^{n}_{1}\right]) \tag{3.23}$$

$$\rightarrow h(\mathbf{s}) - h(\mathbf{u}_{\Lambda(1),\rho(\cdot,1)}) - \sum_{j=2}^{L} \Pr[\boldsymbol{q} \geq j] \cdot \log |[\Lambda(j)/\Lambda(j-1)]|. \tag{3.24}$$

[7] For "simple" distortion measures this definition is sufficient and Voronoi regions are unique. For other distortion measures there may exist non-negligible regions of space which are "closest" to multiple lattice points. In such cases, a tie-breaking rule (*e.g.*, break ties with Euclidean distance) or some other method of specifying Voronoi regions may be necessary. These subtleties are beyond the scope of this work and hence we use the upcoming Definition 2 to restrict attention to cases where these issues do not arise or can be straightforwardly dealt with.

*Proof.* Using the linearity of expectations followed by the law of iterated expectations yields

$$E\left[\frac{1}{n}\sum_{i=1}^{n} d(\mathbf{s}[i], \hat{\mathbf{s}}[i]; \mathsf{q}[i])\right] = \frac{1}{n}\sum_{i=1}^{n} E\left[d(\mathbf{s}[i], \hat{\mathbf{s}}[i]; \mathsf{q}[i])\right] \tag{3.25}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{L} p_{\mathsf{q}}(j) \cdot E\left[d(\mathbf{s}[i], \hat{\mathbf{s}}[i]; \mathsf{q}[i]) | \mathsf{q}[i] = j\right]. \tag{3.26}$$

Combining the i.i.d. nature of the probability models in (3.18) and (3.19), the special form of the distortion measure in (3.20), and the admissibility assumption in Definition 2 establishes that (3.26) implies (3.22).

Next, we have

$$\frac{1}{n}H(\hat{\mathbf{s}}\left[{}^{n}_{1}\right] | \mathsf{q}\left[{}^{n}_{1}\right]) = H(\hat{\mathbf{s}}|\mathsf{q}) \tag{3.27}$$

$$= I(\hat{\mathbf{s}}; \mathbf{s}|\mathsf{q}) \tag{3.28}$$

$$= h(\mathbf{s}|\mathsf{q}) - h(\mathbf{s}|\mathsf{q}, \hat{\mathbf{s}}) \tag{3.29}$$

$$= h(\mathbf{s}|\mathsf{q}) - h(\mathbf{s} - \hat{\mathbf{s}}|\mathsf{q}, \hat{\mathbf{s}}) \tag{3.30}$$

$$\rightarrow h(\mathbf{s}|\mathsf{q}) - h(\mathbf{u}_{\Lambda(\mathsf{q}),\rho(\cdot,\mathsf{q})}|\mathsf{q}) \tag{3.31}$$

$$= h(\mathbf{s}) - h(\mathbf{u}_{\Lambda(\mathsf{q}),\rho(\cdot,\mathsf{q})}|\mathsf{q}) \tag{3.32}$$

$$= h(\mathbf{s}) - \sum_{q=1}^{L} h(\mathbf{u}_{\Lambda(q),\rho(\cdot,q)}) \cdot p_{\mathsf{q}}(q) \tag{3.33}$$

where (3.27) follows from the i.i.d. probability models in (3.18) and (3.19), (3.28) follows from the fact that $\hat{\mathbf{s}}[i]$ is a deterministic function of $\mathbf{s}[i]$ conditioned on $\mathsf{q}[i]$, (3.31) follows from Definition 2, and (3.32) follows from our probability model in which $\mathbf{s}$ and $\mathsf{q}$ are statistically independent.

The random variable $\mathbf{u}_{\Lambda(q),\rho(\cdot,q)}$ is uniform over the Voronoi region $\mathcal{V}[\Lambda(q), \rho(\cdot, q)]$, therefore $h(\mathbf{u}_{\Lambda(q),\rho(\cdot,q)})$ is simply the logarithm of the volume of $\mathcal{V}[\Lambda(q), \rho(\cdot, q)]$. Since $\Lambda(q) \subset \Lambda(q+1)$, however, the Voronoi region of the former is composed of $|[\Lambda(q+1)/\Lambda(q)/|]$ copies of the latter, *i.e.*,

$$\text{Volume}(\mathcal{V}[\Lambda(q), \rho(\cdot, q)]) = |[\Lambda(q+1)/\Lambda(q)]| \cdot \text{Volume}(\mathcal{V}[\Lambda(q+1), \rho(\cdot, q+1)]). \tag{3.34}$$

Repeatedly applying (3.34) and taking the logarithm of the resulting volume yields

$$h(\mathbf{u}_{\Lambda(q),\rho(\cdot,q)}) = h(\mathbf{u}_{\Lambda(1),\rho(\cdot,q)}) + \sum_{j=2}^{q} \log|[\Lambda(j)/\Lambda(j-1)]|. \tag{3.35}$$

Furthermore, for admissible lattices and distortion measures satisfying Definition 2, the volume of the Voronoi region does not depend on the distortion measure [41] and thus we can replace (3.35) with

$$h(\mathbf{u}_{\Lambda(q),\rho(\cdot,q)}) = h(\mathbf{u}_{\Lambda(1),\rho(\cdot,1)}) + \sum_{j=2}^{q} \log|[\Lambda(j)/\Lambda(j-1)]|. \tag{3.36}$$

Substituting (3.36) into (3.33) and collecting terms completes the proof.

□

### 3.5.6 Rate-Distortion For Partially Informed Systems

**Theorem 13.** *For an admissible source, nested lattice, and distortion measure, the average rate and distortion for a partially informed system asymptotically approach the rate and distortion for a fully informed system.*

*Proof.* For samples where $\mathbf{q}[i] = L$, both the partially informed and fully informed system quantize $\mathbf{s}[i]$ to the nearest point in the finest lattice $\Lambda(L)$ and hence achieve the same distortion. For $\mathbf{q}[i] = L$, the fully informed system quantizes $\mathbf{s}[i]$ to $\Lambda(L-1)$. For the partially informed system, $\mathbf{s}[i]$ is quantized to the lattice $\Lambda(L-1)$ shifted by $c[i]$. But this shift does not affect the admissibility assumption in Section 2, and the quantization error for the shifted lattice is still uniformly distributed over the Voronoi region. Thus, the fully informed and partially informed systems asymptotically achieve the same distortion for samples with $\mathbf{q}[i] = L-1$. This argument can be extended to all possible values of $q$ and hence shows that the total average distortion for the fully and partially informed systems are asymptotically equal.

We compute the contributions to the rate from the components of the partially informed system listed in Section 3.4.3. As discussed in (3.10) of Section 3.4.3, the number of bits required to encode $\#[q\left[\begin{smallmatrix}n\\1\end{smallmatrix}\right] \sim j]$ is at most logarithmic in $n$ and hence asymptotically negligible. The rate required to encode the base lattice points $\lambda_{\Lambda(1)}(i)$ is

$$h(\mathbf{s}) - h(\mathbf{u}_{\Lambda(1),\rho(\cdot,1)}) \tag{3.37}$$

by the same arguments as in the proof of Theorem 12. All that remains is to compute the rate required to encode the coset representatives as described in Section 3.4.3.

If an $(n, \#[q\left[\begin{smallmatrix}n\\1\end{smallmatrix}\right] \sim j], n - \#[q\left[\begin{smallmatrix}n\\1\end{smallmatrix}\right] \sim j])$ MDS erasure quantizing code is used to encode the sequence $c[i,j]$ for each $j$, then the total average rate per sample required is

$$\sum_{j=2}^{L} \frac{\#[\mathbf{q}\left[\begin{smallmatrix}n\\1\end{smallmatrix}\right] \sim j]}{n} \cdot \log|[\Lambda(j)/\Lambda(j-1)]|. \tag{3.38}$$

By the law of large numbers, $\#[\mathbf{q}\left[\begin{smallmatrix}n\\1\end{smallmatrix}\right] \sim j]/n$ converges to $p_{\mathbf{q}}(j)$ and thus (3.38) converges to

$$\sum_{j=2}^{L} p_{\mathbf{q}}(j) \cdot \log|[\Lambda(j)/\Lambda(j-1)]|. \tag{3.39}$$

Of course, MDS erasure quantizing codes may not exist for large $n$ or their encoding/decoding may be too complex. Therefore we proposed using near-MDS erasure quantizing codes with parameters $(n, \#[\mathbf{q} \sim j] + \epsilon(n), n - \#[\mathbf{q} \sim j])$ where $\epsilon(n) \to 0$ as $n \to \infty$ (*i.e.*, the asymptotic rate penalty for using near-MDS codes instead of MDS codes is negligible).

Summing the rate components from (3.10), (3.37), and (3.39) shows that the rate for the partially informed system is asymptotically the same as the rate for the fully informed system. □

## ■ 3.6 Concluding Remarks

In this chapter we have described two lattice based quantizers to use distortion side information. The first is a fully informed system where both the encoder and decoder know the distortion side information. It is a natural application of lattice codes and entropy coding to lossy compression with distortion side information and is introduced mainly for the purpose of comparison. The second quantizer is a partially informed system where only the encoder knows the distortion side information. Our main result is that the partially informed system can asymptotically achieve the same performance as the fully informed system and requires only linear complexity.

We feel these results are important for two main reasons. First, the distortion side information framework provides an analytically tractable way to model a rich variety of effects not captured by the traditional distortion measures. As mentioned previously, some examples include perceptual distortion measures and sensor applications. Second, the constructions proposed in this chapter provide a low complexity way to obtain the theoretical benefits of using sophisticated distortion measures in a practical way.

Chapter 4

# Iterative Quantization Using Codes On Graphs

In this chapter, we study codes on graphs combined with an iterative message passing algorithm for quantization. Specifically, we consider the binary erasure quantization (BEQ) problem which is the dual of the binary erasure channel (BEC) coding problem. We show that duals of capacity achieving codes for the BEC yield codes which approach the minimum possible rate for the BEQ. In contrast, low density parity check codes cannot achieve the minimum rate unless their density grows at least logarithmically with block length. Furthermore, we show that duals of efficient iterative decoding algorithms for the BEC yield efficient encoding algorithms for the BEQ. Hence our results suggest that graphical models may yield near optimal codes in source coding as well as in channel coding and that duality plays a key role in such constructions.

## ■ 4.1 Introduction

Researchers have discovered that error correction codes defined on sparse graphs can be iteratively decoded with low complexity and vanishing error probability at rates close to the Shannon limit. Based on the close parallels between error correction and data compression, we believe that similar graphical codes can approach the fundamental limits of data compression with reasonable complexity. Unfortunately, the existing suboptimal channel decoding algorithms for graphical codes generally fail unless the decoder input is already near a codeword. Since this is usually not the case in source coding, either a new type of graph or a new suboptimal algorithm (or both) is required.

Before developing iterative quantization techniques it is worth investigating the potential gains of such an approach over existing systems. For asymptotically high rates, when compressing a continuous source with finite moments relative to mean square error (MSE) distortion, entropy coded scalar quantization (ECSQ) is 1.53 dB from the rate-distortion limit [79]. For moderate rates the gap is larger: in quantizing a Gaussian source relative to MSE distortion, ECSQ systems are 1.6–3.4 dB away from the rate-distortion limit. For these parameters, trellis coded quantization (TCQ) using a 256-state code with optimal quantization has a gap of 0.5–1.4 dB [117]. For higher rates, sources with larger tails (*e.g.*, a source with a Laplacian distribution), or sources with memory, the gaps are larger. Thus for memoryless sources quantized at moderate rates, new codes have the potential to improve performance by the noticeable margin of a few decibels. More generally, the codes on graphs paradigm may prove valuable in a variety of scenarios involving speech, audio, video and other complicated sources.

To illustrate possible approaches to developing graphical codes we focus on the binary erasure quantization (BEQ) problem which is the dual of the binary erasure channel (BEC) coding problem. First we describe the BEQ problem model in Section 4.2. Next, in Section 4.3 we present our main result for the BEQ: duals of low density parity check codes can be analyzed, encoded, and decoded by dualizing the corresponding techniques for the BEC. Specifically, by dualizing capacity achieving codes for the BEC we obtain rate-distortion approaching codes for the BEQ. Finally, we close with some concluding remarks in Section 4.4.

## ■ 4.2 Quantization Model

Vectors and sequences are denoted with an arrow (*e.g.*, $\mathbf{x}$). Random variables or random vectors are denoted using the sans serif font (*e.g.*, $\mathsf{x}$ or $\boldsymbol{\mathsf{x}}$). We consider the standard (memoryless) data compression problem and represent an instance of the problem with the tuple $(\mathcal{S}, p_{\mathsf{s}}(s), d(\cdot,\cdot,))$ where $\mathcal{S}$ represents the source alphabet, $p_{\mathsf{s}}(s)$ represents the source distribution, and $d(\cdot,\cdot,)$ represents a distortion measure. Specifically, a source $\boldsymbol{\mathsf{s}}$ consists of a sequence of $n$ random variables $\mathsf{s}_1$, $\mathsf{s}_2$, …, $\mathsf{s}_n$ each taking values in $\mathcal{S}$ and generated according to the distribution $p_{\boldsymbol{\mathsf{s}}}(\mathbf{s}) = \prod_{i=1}^{n} p_{\mathsf{s}}(s_i)$. A rate $R$ encoder $f\cdot$ maps $\boldsymbol{\mathsf{s}}$ to an integer in $\{1, 2, \ldots, 2^{nR}\}$, and the corresponding decoder $g\cdot$ maps the resulting integer into a reconstruction $\hat{\boldsymbol{\mathsf{s}}}$. Distortion between the source $\boldsymbol{\mathsf{s}}$ and the reconstruction $\hat{\boldsymbol{\mathsf{s}}}$ is measured via $D = \frac{1}{n}\sum_{i=1}^{n} d(\mathsf{s}_i, \hat{\mathsf{s}}_i,)$.

Shannon derived the minimum possible rate required by any data compression system operating with distortion $D$. The so-called rate-distortion function is given by the formula

$$R(D) = \min_{p_{\hat{\mathsf{s}}|\mathsf{s}}(\hat{s}|s):E[d(\mathsf{s},\hat{\mathsf{s}},])\leq D} I(\mathsf{s};\hat{\mathsf{s}}) \tag{4.1}$$

where $I(\cdot;\cdot)$ denotes mutual information and $E[\cdot]$ denotes expectation.

### 4.2.1 Binary Erasure Quantization

To highlight connections between error correction and data compression, we consider the binary erasure quantization (BEQ) problem where the source vector consists of ones, zeros, and "erasures" represented by the symbol $*$. Neither ones nor zeros may be changed, but erasures may be quantized to either zero or one. Practically, erasures may represent source samples which are missing, irrelevant, or corrupted by noise and so do not affect the distortion regardless of the value they are assigned. Formally, the BEQ problem with erasure probability $e$ corresponds to

$$\mathcal{S} = \{0, 1, *\} \tag{4.2a}$$

$$p_{\mathsf{s}}(s) = \frac{1-e}{2}\cdot\delta(s) + \frac{1-e}{2}\cdot\delta(s-1) + e\cdot\delta(s-*) \tag{4.2b}$$

$$d(a, b,) = 0 \text{ if } a = * \text{ or } a = b, \text{ and 1 otherwise.} \tag{4.2c}$$

It is straightforward to show that for $D = 0$ the distribution

$$p_{\hat{\mathsf{s}}|\mathsf{s}}(\hat{s}|s) = \delta(\hat{s} - s) \text{ if } s \in \{0, 1\}, \text{ and } \frac{1}{2}\cdot\delta(\hat{s}) + \frac{1}{2}\cdot\delta(\hat{s}-1) \text{ if } s = *. \tag{4.3}$$

optimizes (4.1) and yields the value of the rate-distortion function at $D = 0$:

$$R_{\text{BEQ}}(D = 0) = 1 - e. \tag{4.4}$$

## 4.3 Codes For Erasure Quantization

It is well-known that the encoder for a quantizer serves a similar function to the decoder for an error correcting code in the sense that both take a vector input (*i.e.*, a source to quantize or channel output to decode) and map the result to bits (*i.e.*, the compressed source or the transmitted message). The decoder for a quantizer can similarly be identified with the encoder for an error correcting code in the sense that both take bits as input and produce a vector (*i.e.*, a source reconstruction or a channel input). Thus it is natural to investigate whether swapping the encoder and decoder for a good error correcting code such as a low density parity check (LDPC) code produces a good quantizer.

### ■ 4.3.1 LDPC Codes Are Bad Quantizers

One benefit of studying the BEQ problem is that it demonstrates why low density parity check (LDPC) codes are inherently unsuitable for quantization. Specifically, consider an LDPC code like the one illustrated in Fig. 4-1 using Forney's normal graph notation [66]. If all the variables connected to a given check are not erased, then there is an even chance that no code symbol can match the source in that position and thus the distortion will be positive regardless of the code rate. Thus, as stated in Theorem 14 and proved in Appendix C, successful decoding is asymptotically unlikely unless the density of *every* parity check matrix for the code increases logarithmically with the block length.[1]

**Theorem 14.** *Let $\mathcal{C}_{(n)}$ be a sequence of linear codes of length $n$ and fixed rate $R$ such that the probability that binary erasure quantization using $\mathcal{C}_{(n)}$ of a random source sequence with $e \cdot n$ erasures will succeed with zero distortion is bounded away from 0 as $n \to \infty$. Then regardless of the values of $R$ and $e$, the degree of the parity-check nodes in any parity-check graph representation of $\mathcal{C}_{(n)}$ must increase at least logarithmically with $n$.*

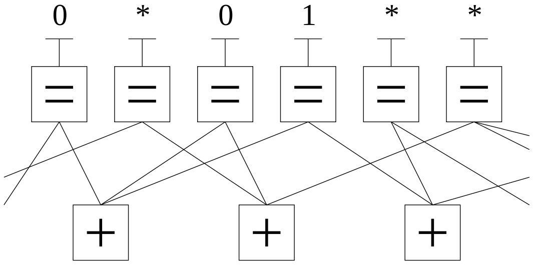


**Figure 4-1.** Using an LDPC code for binary erasure quantization. The boxes with = signs are repetition nodes: all edges connected to an = box must have the same value. The boxes with + signs are check nodes: the modulo 2 sum of the values on edges connected to a + box must be 0. The source consists of 0's, 1's, and erasures represented by $*$'s. Erasures may be quantized to 0 or 1 while incurring no distortion. A non-zero distortion must be incurred for the source shown above since the left-most check cannot be satisfied.

The poor performance of LDPC codes for quantization may seem surprising in light of their excellent properties in channel coding, but it has long been recognized that good codes for error correction and quantization may be different. The former is essentially a packing problem where the goal is to place as many codewords as possible in a given space such that the codewords are far apart and can be distinguished despite noise. The latter is a covering problem where the goal is to place as few codewords as possible in a given space such that every point in space is near at least one codeword.

From any good error correcting code (respectively data compression code) it is easy to obtain another code which is almost as good at error correction (resp. data compression) but terrible at source coding (resp. channel coding). For example, removing half the codewords has an asymptotically negligible effect on error correction since it only decreases the rate by $1/n$ and only increases robustness. But, removing half the codewords can dramatically hinder source coding since half the time the source may be very far from the nearest codeword. Conversely, doubling the number of codewords has an asymptotically negligible effect on data compression since the rate only increases by $1/n$ while the distortion

[1] We may expect the density of a code to increase as it approaches the *capacity* or *rate-distortion function*, but a code whose density also increases with *block length* seems undesirable.

may decrease slightly. But doubling the number of codewords can be catastrophic for error correction if it drastically reduces the distance between codewords.

## ■ 4.3.2 Dual LDPC Codes

Many researchers have explored duality relationships between error correction and source coding. Such work demonstrates that often a good solution for one problem can be obtained by dualizing a good solution to the other. Continuing in this tradition, we study the properties of dual LDPC codes for binary erasure quantization.

Formally, a length $n$ binary linear code $\mathcal{C}$ is a subspace of the $n$ dimensional vector space over the binary field and the dual code $\mathcal{C}^{\perp}$ is the subspace orthogonal to $\mathcal{C}$. For LDPC codes, the code $\mathcal{C}$ is usually specified by the parity check matrix $\mathbf{H}$ representing the constraint that $\mathbf{H}\mathbf{x}^T = 0$ if and only if $\mathbf{x}$ is a codeword. To obtain the dual code $\mathcal{C}^{\perp}$ we can recall that the generator $\mathbf{G}^{\perp}$ of $\mathcal{C}^{\perp}$ is exactly $\mathbf{H}$. If the code $\mathcal{C}$ is represented by a normal graph as in Fig. 4-1, then the graph of the dual code $\mathcal{C}$ can be obtained by swapping $+$ and $=$ nodes [66]. In dualizing the code graph in this manner it may be useful to note that while the graph of $\mathcal{C}$ obtained from $\mathbf{H}$ represents a syndrome former for $\mathcal{C}$, the dualized graph represents an encoder for $\mathcal{C}^{\perp}$.

For example, Fig. 4-2 is obtained by dualizing the code graph in Fig. 4-1. Notice that while the original code cannot quantize the source with distortion 0, the dual code can. Intuitively, the advantage of a low density encoder structure is that it provides a simple representation of a basis which can be used to construct the desired vector. In the following sections, we investigate the properties of dual LDPC codes for quantization with both optimal quantization and iterative quantization.

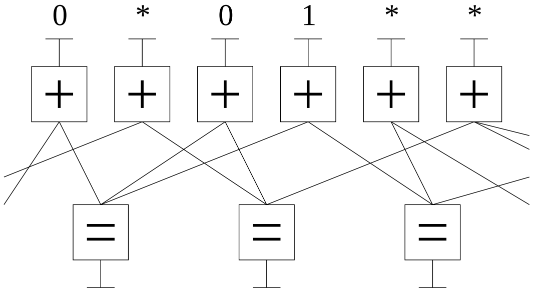


**Figure 4-2.** Using the dual of an LDPC code for binary erasure quantization. Choosing values for the variables at the bottom produces a codeword. The values for each sample of the resulting codeword are obtained by taking the sum modulo 2 of the connected variables. In contrast to Fig. 4-1, this structure can successfully match the source with no distortion if the bottom 3 variables are set to 0, 0, 1.

## ■ 4.3.3 Optimal Quantization/Decoding and Duality

The following theorem (proved in Appendix C) demonstrates the dual relationship between channel decoding and source quantization using optimal decoding/quantization algorithms.

**Theorem 15.** *A channel decoder for the code $\mathcal{C}$ can correctly decode every received sequence with the erasure pattern*[2] $\mathbf{e}$ *if and only if a quantizer for the code $\mathcal{C}^{\perp}$ can successfully quantize*

[2] If symbol $i$ is erased (resp. unerased) then $e_i = 1$ (resp. $e_i = 0$) in our notation for erasure patterns.

*every*[3] *source sequence with the erasure pattern* $\mathbf{e}^{\perp} = 1 - \mathbf{e}$.

From this result we immediately obtain the following Corollary.

**Corollary 3.** *Let* $\mathcal{C}_{(n)}$ *be a sequence of linear codes which achieves the capacity of a binary erasure channel with erasure probability* $e$ *using optimal decoding. The sequence* $\mathcal{C}_{(n)}^{\perp}$ *obtained by taking the duals of* $\mathcal{C}_{(n)}$ *achieves the minimum rate for* $D = 0$ *for the BEQ with erasure probability* $e^{\perp} = 1 - e$ *using optimal quantization.*

The statement and proof of the two preceding results contain a curious duality between erased/known symbols in source coding and known/erased symbols in channel coding. A similar duality exists between a likelihood ratio, $\lambda$, and its Fourier transform $\Lambda = \frac{1-\lambda}{1+\lambda}$ used in dualizing the sum-product algorithm [66, pp. 545–546]. Specifically, the Fourier transform maps known/erased likelihood ratios to erased/known likelihood ratios.

## 4.3.4 Iterative Decoding/Quantization and Duality

In the following we first review the intuition behind iterative erasure decoding algorithms and describe the particular decoding algorithm we consider in Table 4.1. Next we outline the intuition behind a similar approach for iterative quantization and precisely describe our quantization algorithm in Table 4.2. Finally, we show that these algorithms are duals.

### Iterative Erasure Decoding

Many iterative message-passing decoding algorithms are essentially based on the following idea. The outgoing message on edge $i$ of a $+$ node is the modulo-2 sum of all incoming messages (excluding edge $i$) with the proviso that if any incoming message (excluding edge $i$) is $*$ then the outgoing message is also $*$. For an $=$ node, the outgoing message on edge $i$ is $*$ only if all other incoming messages are $*$, otherwise the outgoing message is the same as the known incoming message or messages. These message-passing rules can be interpreted as determining the outgoing message on edge $i$ by applying the following "sum" and "product" formulas to all other incoming messages. [4]

$$\begin{array}{c||c|c|c} + & 0 & 1 & * \\ \hline\hline 0 & 0 & 1 & * \\ \hline 1 & 1 & 0 & * \\ \hline * & * & * & * \end{array} \quad \text{and} \quad \begin{array}{c||c|c|c} \times & 0 & 1 & * \\ \hline\hline 0 & 0 & \# & 0 \\ \hline 1 & \# & 1 & 1 \\ \hline * & 0 & 1 & * \end{array} \tag{4.5}$$

It is well-known that such algorithms yield optimal decoding on a tree and also perform well on graphs with cycles provided appropriate scheduling and initialization rules are selected. Initializing all messages to $*$ and using sequential or parallel schedules are common choices. For the purpose of proving theorems, we consider a sequential schedule in the ERASURE-DECODE algorithm of Table 4.1.

[3] Note that some source sequences (*e.g.*, the all zero sequence) can be successfully quantized using $\mathcal{C}^{\perp}$ regardless of $\mathbf{e}^{\perp}$. Similarly, a system which decodes ambiguous received sequences to the all zero sequence may succeed even when many erasures occur. Thus to obtain the desired equivalence between correct decoding and successful quantization we define correct decoding (resp. successful quantization) as being able to deduce the transmitted codeword (resp. a codeword matching non-erased positions of the source) for every possible received sequence (resp. source) with the the erasure pattern $\mathbf{e}$ (resp. $\mathbf{e}^{\perp}$).

[4] In the product rule for erasure decoding, the symbol # denotes a contradiction which is impossible if only erasures and no errors occurred.

**Table 4.1.** An algorithm for iteratively decoding data with erasures.

```
ERASURE-DECODE($\mathbf{H}$,$\mathbf{y}$)
while $\mathbf{y}$ has at least one erased sample do
  if $\exists$ row $i$ of $\mathbf{H}$ (i.e., a check) connected to exactly one erased variable $y_j$ then
    Set $y_j$ to be the XOR of all unerased bits in the check
  else
    return FAIL
  end if
end while
Set $\mathbf{x}$ to the message variables obtained from $\mathbf{y}$
return the message variables $\mathbf{x}$
```

### Iterative Erasure Quantization

The message-passing rules in (4.5) can also be applied to the BEQ problem for graphs without cycles provided some form of tie-breaking is used. Specifically, some variables will receive erasure messages even after the algorithm has completed. Such variables can be arbitrarily chosen to be either 0 or 1 and still produce a valid quantization. For example in quantizing the source $(\ast, \ast, 1)$ with a (3,2) single parity check code, both $(1, 0, 1)$ and $(0, 1, 1)$ are equally valid results and this tie can broken arbitrarily.

On a graph with cycles, however, generalizing this approach by initializing all unknown messages to $\ast$ usually fails. For example, on the dual of a Gallager code or a code like the one represented in Fig. 4-2 such an initialization rule leads to all messages being erased at every step of the algorithm. To perform effective tie-breaking, we need to distinguish between variables which can be arbitrarily set to 0 or 1 and variables which have not yet received enough information to be determined.

One way to distinguish between these cases is to denote the former as erasures with the symbol $\ast$ and the latter as null messages with the symbol $\varnothing$, and initialize all messages to $\varnothing$. With this initialization, we can use the following "sum" and "product" rules: [5]

$$\begin{array}{c||c|c|c|c}
+ & 0 & 1 & \ast & \varnothing \\
\hline\hline
0 & 0 & 1 & \ast & \varnothing \\
\hline
1 & 1 & 0 & \ast & \varnothing \\
\hline
\ast & \ast & \ast & \ast & \ast \\
\hline
\varnothing & \varnothing & \varnothing & \ast & \varnothing \\
\hline
\end{array}
\quad \text{and} \quad
\begin{array}{c||c|c|c|c}
\times & 0 & 1 & \ast & \varnothing \\
\hline\hline
0 & 0 & \# & 0 & 0 \\
\hline
1 & \# & 1 & 1 & 1 \\
\hline
\ast & 0 & 1 & \ast & \varnothing \\
\hline
\varnothing & 0 & 1 & \varnothing & \varnothing \\
\hline
\end{array}
\tag{4.6}$$

Specifically, the outgoing message from a + node in a graph like Fig. 4-2 is computed by combining incoming messages from all other edges with the + rule. The outgoing message from an = node is computed by combining incoming messages from all other edges with the × rule. Whenever an = node has all incoming messages being $\ast$, the value of the node is arbitrary. This tie can be broken by arbitrarily choosing a value of 0 or 1 provided the tie is broken consistently. Essentially, the requirement of consistent tie-breaking can be interpreted as a constraint on the message-passing schedule: tie-breaking information for a given tie should be propagated through the graph before other ties are broken.

In order to provide a precise algorithm for the purpose of proving theorems, we consider the ERASURE-QUANTIZE in Table 4.2 based on applying the rules in (4.6) with a sequential

[5] In the product rule for erasure quantization, the symbol # denotes a contradiction. If a contradiction is generated then quantizing the given source with no distortion is impossible and the algorithm fails.

schedule and all tie-breaking collected into step 8.

**Table 4.2.** An algorithm for iteratively quantizing a source with erasures.

```
ERASURE-QUANTIZE($\mathbf{G}$,$\mathbf{z}$)
 1: while $\mathbf{z}$ has at least one unerased sample do
 2:   if $\exists$ row $i$ of $\mathbf{G}$ (i.e., a variable) connected to exactly one unerased check $z_j$ then
 3:     Reserve message variable $i$ to later satisfy $z_j$ and erase check $z_j$
 4:   else
 5:     return FAIL
 6:   end if
 7: end while
 8: Arbitrarily set all unreserved message variables
 9: Set reserved variables to satisfy the corresponding checks starting from the last reserved variable
    and working backward to the first reserved variable
10: return message variables $\mathbf{w}$
```

### Iterative Algorithm Duality

Our main results regarding iterative quantization are the following three theorems stating that ERASURE-QUANTIZE works correctly, can be analyzed in the same manner as the ERASURE-DECODE algorithm, and works quickly:

**Theorem 16.** *For any linear code with generator matrix* $\mathbf{G}$*,* ERASURE-QUANTIZE($\mathbf{G}, \mathbf{z}$) *either fails in step 5 or else returns* $\mathbf{w}$ *such that* $\mathbf{wG}$ *matches* $\mathbf{z}$ *in all unerased positions.*

**Theorem 17.** *Consider a linear code with parity check matrix* $\mathbf{H}$ *and its dual code with generator matrix* $\mathbf{G}^{\perp} = \mathbf{H}$*. The algorithm* ERASURE-DECODE($\mathbf{H}, \mathbf{y}$) *fails in step 5 if and only if the algorithm* ERASURE-QUANTIZE($\mathbf{G}^{\perp}, \mathbf{z}$) *fails in step 5 where* $\mathbf{y}$ *has erasures specified by* $\mathbf{e}$ *and* $\mathbf{z}$ *has erasures specified by* $\mathbf{e}^{\perp} = 1 - \mathbf{e}$*.*

**Theorem 18.** *The algorithm* ERASURE-QUANTIZE($\mathbf{G}, \mathbf{z}$) *runs in time* $\mathcal{O}(n \cdot d)$ *where* $n$ *is the length of* $\mathbf{z}$ *and* $d$ *is the maximum degree of the graph corresponding to* $\mathbf{G}$*.*

These results (proved in Appendix C) imply that the parallel structure between erasure decoding and erasure quantization allows us to directly apply virtually every result from the analysis of one to the other. For example, these theorems combined with the analysis/design of irregular LDPC codes achieving the capacity of the binary erasure channel [139] immediately yield the following Corollary:

**Corollary 4.** *There exists a sequence of linear codes which can be efficiently encoded and decoded that achieves the rate-distortion function for binary erasure quantization.*

## ■ 4.4 Concluding Remarks

In this chapter we demonstrated how codes on sparse graphs combined with iterative decoding can achieve the Shannon limit for binary erasure quantization. The main contribution of our algorithm is in recognizing the role of tie-breaking, scheduling, and initialization in iterative quantization. The key insight in our analysis is the strong dual relationship between error correction and quantization for codes on graphs and their associated decoding/quantization algorithms (both optimal and iterative). We conjecture that the main task in designing iterative message-passing algorithms for more general quantization problems lies in designing appropriate tie-breaking, scheduling, and initialization rules for such scenarios and exploiting similar dual relationships to channel decoding.

Chapter 5

# Source Coding with Fixed Lag Side Information

## ■ 5.1 Introduction

There is a growing consensus that understanding complex, distributed systems requires a combination of ideas from communication and control [132]. Adding communication constraints to traditional control problems or adding real-time constraints to communication problems has recently yielded interesting results [30,55,157,169,188]. We consider a related aspect of this interaction by exploring the possible advantages that the feedback/feedforward in control scenarios can provide in compression. Specifically, we explore a variant of the Wyner-Ziv problem [189] where causal side information about the source is available with a fixed lag to the decoder and explore how such side information may be used.

For example, consider a remote sensor that sends its observations to a controller as illustrated in Fig. 5-1. The sensor may be a satellite or aircraft reporting the upcoming temperature, wind speed, or other weather data to a vehicle. The sensor observations must be encoded via lossy compression to conserve power or bandwidth. In contrast to the standard lossy compression scenario, however, the controller directly observes the original, uncompressed data after some delay. The goal of the sensor observations is to provide the controller with information about upcoming events *before* they occur. Thus at first it might not seem that observing the true, uncompressed data *after* they occur would be useful. Our main goal is to try to understand how these delayed observations of the source data (which we call side information) can be used. Our main result is that such information can be quite valuable.

The following toy problem helps illustrate some relevant issues. Imagine that Alice plays a game where she will be asked 10 Yes/No questions. Of these questions, 5 have major prizes while the others have minor prizes. After answering each question, she is told the correct answer as well as what the prize for that question is and receives the prize if she is correct. Bob knows all the questions and the corresponding prizes beforehand and wishes to help Alice by preparing a "cheat-sheet" for her. But Bob only has room to record 5 answers. Is there a cheat-sheet encoding strategy that guarantees that Alice will always correctly answer the questions with the 5 best prizes? No such strategy exists using a classical compression scheme. Instead, as illustrated in Section 5.3, the optimal strategy requires an encoding which uses the fact that Alice gains information about the prize *after* answering.

In the rest of the chapter, we study the fixed lag side information problem. Since solving the general problem seems difficult, we begin by focusing on perfect side information with a unit lag. This special case is the feedforward source coding problem introduced by Pradhan [144] and is dual (in the sense of [145, 16]) to channel coding with feedback. By using the feedforward side information, it is possible to construct low complexity source

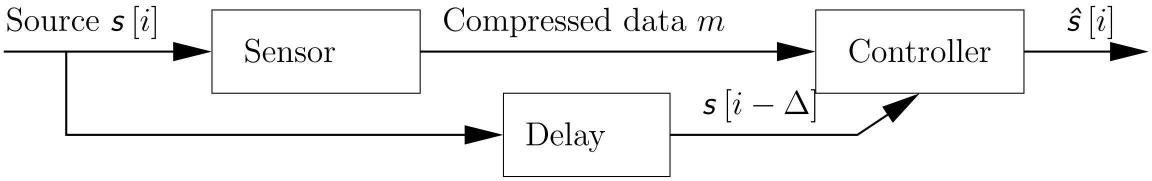


**Figure 5-1.** A sensor compresses and sends the source sequence $\mathsf{s}[1]$, $\mathsf{s}[2]$, …, to a controller which reconstructs the quantized sequence $\hat{\mathsf{s}}[1]$, $\hat{\mathsf{s}}[2]$, …, in order to take some control action. After a delay or lag of $\Delta$, the controller observes the original, uncompressed data directly.

coding systems which can achieve the rate-distortion bound. Specifically, [144] describes how to adapt the Kailath-Schalkwijk scheme for the Gaussian channel with feedback [160] to the Gaussian source squared distortion scenario with feedforward side information. In this chapter, we consider finite alphabet sources with arbitrary memoryless distributions and arbitrary distortion measures. Since Ooi and Wornell's channel coding with feedback scheme [137] achieves the best error exponent with minimum complexity, we investigate the source coding dual. Specifically, we show that the source coding dual of the Ooi–Wornell scheme achieves the rate-distortion bound with linear complexity.

We begin by describing the problem in Section 5.2. Next we present a simple example of how feedforward side information can be useful in the binary erasure quantization problem in Section 5.3. In Section 5.4, we consider the more complicated example of quantizing a binary source with respect to Hamming distortion. We present our source coding algorithm for general sources in Section 5.5 and show that it achieves the rate-distortion function with low complexity. We close with some concluding remarks in Section 5.6.

## ■ 5.2 Problem Description

Random variables are denoted using the sans serif font (*e.g.*, $\mathsf{x}$) with deterministic values using serif fonts (*e.g.*, $x$). We represent the $i$th element of a sequence as $x\left[i\right]$ and denote a subsequence including elements from $i$ to $j$ as $x\left[\genfrac{}{}{0pt}{}{j}{i}\right]$.

We consider (memoryless) source coding with fixed lag side information and represent an instance of the problem with the tuple $(\mathcal{S}, \mathcal{W}, p_{\mathsf{s},\mathsf{w}}, d(\cdot,\cdot,,)\Delta)$ where $\mathcal{S}$ and $\mathcal{W}$ represent the source and side information alphabets, $p_{\mathsf{s},\mathsf{w}}(s,w)$ represents the source and side information joint distribution, $d(\cdot,\cdot,)$ represents the distortion measure, and $\Delta$ represents the delay or lag. Specifically, the source and side information each consist of a sequence of $n$ random variables $\mathsf{s}\left[\genfrac{}{}{0pt}{}{n}{1}\right]$ and $\mathsf{w}\left[\genfrac{}{}{0pt}{}{n}{1}\right]$ taking values in $\mathcal{S}$ and $\mathcal{W}$ generated according to the distribution $p_{\mathsf{s}\left[\genfrac{}{}{0pt}{}{n}{1}\right],\mathsf{w}\left[\genfrac{}{}{0pt}{}{n}{1}\right]}(s\left[\genfrac{}{}{0pt}{}{n}{1}\right]) = \prod_{i=1}^{n} p_{\mathsf{s},\mathsf{w}}(s\left[i\right], w\left[i\right])$.

A rate $R$ encoder, $f(\cdot)$, maps $\mathsf{s}\left[\genfrac{}{}{0pt}{}{n}{1}\right]$ to a bit sequence represented as an integer $m \in \{1, 2, \ldots, 2^{nR}\}$. The corresponding decoder $f^{-1}(\cdot)$ works as follows. At time $i$, the decoder takes as input $m$ as well as the side information samples, $\mathsf{w}\left[\genfrac{}{}{0pt}{}{i-\Delta}{1}\right]$, and produces the $i$th reconstruction $\hat{\mathsf{s}}\left[i\right]$. A distortion of $d(\mathsf{s}\left[i\right], \hat{\mathsf{s}}\left[i\right],)$ is then incurred for the $i$th sample where $d(\cdot,\cdot,)$ is a mapping from $\mathcal{S} \times \mathcal{S}$ to the interval $[0, d_{\max}]$.

The basic problem can be specialized to the original (non-causal) Wyner-Ziv problem [189], by allowing a negative delay $\Delta = -\infty$. Similarly, setting $\Delta = 0$ yields a causal version of the Wyner-Ziv problem. Finally, letting the side information be exactly the same as the source with a positive delay yields the feedforward source coding problem studied in [144]. For all these cases, the goal is to understand the fundamental rate-distortion-complexity performance. To show that the benefits of fixed lag side information are worth investigating, we focus on the feedforward case where $\mathsf{w} = \mathsf{s}$ with unit delay $\Delta = 1$ throughout the rest of this chapter.

For memoryless sources, the information feedforward rate-distortion function, $R_{\mathrm{f}}^{(I)}(D)$, is defined to be the same as Shannon's classical rate-distortion function:

$$R_{\mathrm{f}}^{(I)}(D) = \inf_{p_{\hat{\mathsf{s}}|\mathsf{s}}:E[d(\mathsf{s},\hat{\mathsf{s}},)]\leq D} I(\hat{\mathsf{s}};\mathsf{s}). \tag{5.1}$$

The operational feedforward rate-distortion function, $R_{\mathrm{f}}(D)$, is the minimum rate required such that there exists a sequence of encoders and decoders with average distortion,

$$\frac{1}{n}\sum_{i=1}^{n} d(s[i], \hat{s}[i]),$$

asymptotically approaching $D$. As observed in [144] and shown in the appendix, the information and operational feedforward rate-distortion functions are the same. Thus feedforward does not reduce the rate required, but as we argue in the rest of this chapter, it allows us to approach the rate-distortion function with low complexity.

## ■ 5.3 Example: Binary Source & Erasure Distortion

The simplest example in channel coding with feedback is the erasure channel and in this case the algorithm in Fig. 5-2 achieves capacity. At time 1 the encoder puts message bit $m[1]$ into the channel. If it is received correctly, the transmitter then transmits $m[2]$, otherwise $m[1]$ is repeated until it is successfully received. The same process is used for $m[2]$, $m[3]$, etc. For example, to send the message 0101 though a channel where samples 2, 3, 6, and 7 are erased, the transmitter would send 01110111 and the receiver would see $0**10**1$. In general, if there are $n$ message bits, $m[1]$, $m[2]$, ..., $m[n]$, and $e$ erasures, then exactly $n+e$ channel uses are required. This yields a transmission rate of $n/(n+e)$ which is exactly the channel capacity.

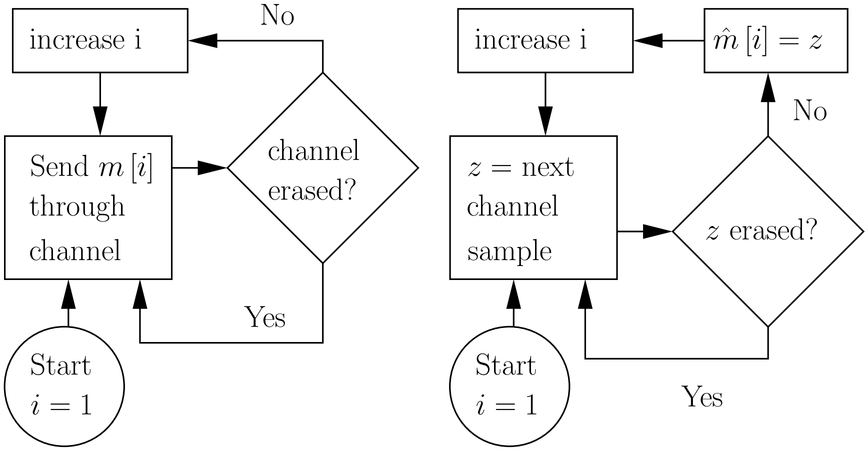


**Figure 5-2.** Encoder (left) for transmitting a message $m = m_1, m_2, \ldots$ across an erasure channel with feedback and decoder (right) for producing an estimate of the transmitted message $\hat{m}$.

The dual to the binary erasure channel (BEC) is the binary erasure quantization problem (BEQ). In the BEQ, each source sample can be either 0, 1, or * where * represents "don't care". The distortion measure is such that 0 and 1 cannot be changed but * can be quantized to either 0 or 1 with no distortion. The BEQ models the game introduced in the introduction.[1] To develop a source coding with feedforward algorithm for the BEQ, we can dualize the channel coding with feedback algorithm for the BEC as illustrated in Fig. 5-3.

[1] Yes/No answers for questions with major prizes map to 1/0 values for the source while questions with minor prizes map to a value of * for the source. The distortion measure represents the restriction that questions with major prizes must be answered correctly while the answers for the other questions are irrelevant.

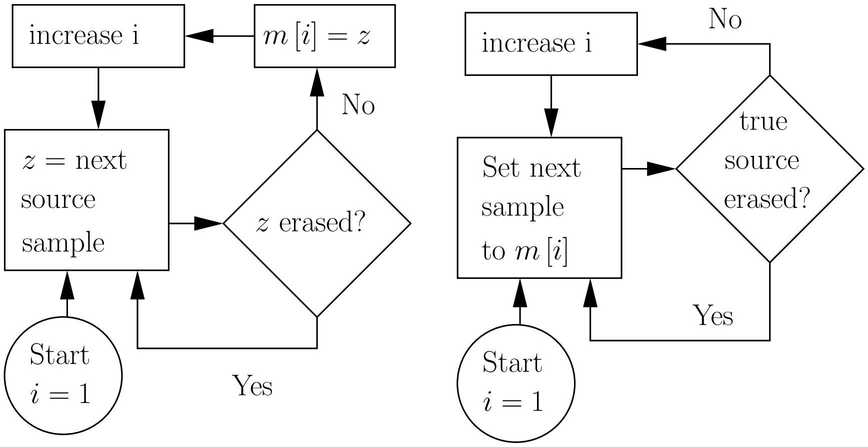


**Figure 5-3.** Encoder (left) and decoder (right) for the binary erasure quantization problem which is dual to the binary erasure channel.

Assume the source is $\mathsf{s}\left[{}^{8}_{1}\right] = 0 * *10 * *1$. The encoder compresses this to $m = 0101$ by ignoring all the * symbols and sends this to the receiver. At time 1, the receiver chooses $\hat{\mathsf{s}}[1]$ to be the first bit in the encoding (i.e., $\hat{\mathsf{s}}[1] = m[1] = 0$). From the feed-forward the receiver realizes this is correct after it makes its choice. Next, the receiver chooses $\hat{\mathsf{s}}[2]$ to be the next bit received (i.e., $\hat{\mathsf{s}}[2] = m[2] = 1$). After choosing this reconstruction, the receiver is told that in fact $\mathsf{s}[2] = *$ so even though $\mathsf{s}[2] \neq \hat{\mathsf{s}}[2]$, no distortion is incurred. At this point, the receiver realizes that $m[2]$ must have been intended to describe something after $\mathsf{s}[2]$. So at time 3, the receiver chooses $\hat{\mathsf{s}}[3] = m[2] = 1$. Once again the receiver learns that this is incorrect since $\mathsf{s}[3] = *$, but again no penalty is incurred. Again, the receiver decides that $m[2]$ must have been intended to describe something else so it chooses $\hat{\mathsf{s}}[4] = m[2] = 1$ at time 4. This turns out to be correct and so at time 5 the receiver chooses $\hat{\mathsf{s}}[5] = m[3]$, etc.

In the encoder/decoder described above, the encoder sends the non-erased bits of $\mathsf{s}\left[{}^{n}_{1}\right]$ and the decoder tries to match up the compressed data to the source. This system yields distortion 0 provided that at least $n - e$ bits are sent where $e$ denotes the number of * symbols in the source vector. It is straightforward to show that no encoder/decoder can do better for any value of $n$ or $e$. A system not taking advantage of the feedforward could asymptotically achieve the same performance but it would require more complexity and more redundancy. Thus just as in the erasure channel with feedback, we see that for the erasure source, feedforward allows us to achieve the minimum possible redundancy with minimum complexity.

## ■ 5.4 Example: Binary Source & Hamming Distortion

The example in Section 5.3 illustrates that feedforward can be useful in source coding by using some special properties of the BEQ problem. Next, we consider a somewhat more complicated example to illustrate that a key idea in developing *lossy* compression algorithms for source coding with feedforward is the use of classical *lossless* compression algorithms. Specifically, we consider a binary source which is equally likely to be either zero or one, and we consider the Hamming distortion measure $d(s, \hat{s}) = |s - \hat{s}|$. As is well known, the rate-distortion function for this case is $R(D) = 1 - H_b(D)$ where $H_b(\cdot)$ is the binary entropy function. In the following we outline a scheme which achieves a distortion of $D_0 = 0.11$ and rate $R(D_0) = 1 - H_b(0.11) \approx 1/2$.

Let $\mathcal{C}\left(\cdot\right)$ and $\mathcal{C}^{-1}\left(\cdot\right)$ be a lossless compression and decompression algorithm for a Bernoulli source with a fraction $D_0$ of ones. Specifically, $\mathcal{C}\left(\cdot\right)$ takes as input $t$ bits with a fraction $D_0$ ones and maps them into $H_b(D_0)t \approx t/2$ uniformly distributed bits while $\mathcal{C}^{-1}\left(\cdot\right)$ maps $t'$ approximately uniformly distributed bits into $t'/H_b(D_0) \approx 2t'$ bits with a fraction $D_0$ ones. To simplify the exposition, we assume that for $t' \geq \mathtt{M}$, these approximations are exact. A more careful treatment appears in Section 5.5.

The feedforward lossy compression system encoder takes a sequence of source samples, $\mathsf{s}\left[\substack{n\\1}\right]$, where $n = \mathtt{M}(2^K - 1)$ and encodes by producing the following codewords:

$$b_1 \triangleq \mathsf{s}\left[\substack{n\\n-\mathtt{M}+1}\right] \tag{5.2}$$

$$b_2 \triangleq \mathsf{s}\left[\substack{n-\mathtt{M}\\n-3\mathtt{M}+1}\right] \oplus \mathcal{C}^{-1}\left(b_1\right) = \mathsf{s}\left[\substack{n-\mathtt{M}\\n-3\mathtt{M}+1}\right] \oplus \mathcal{C}^{-1}\left(\mathsf{s}\left[\substack{n\\n-\mathtt{M}+1}\right]\right) \tag{5.3}$$

$$b_3 \triangleq \mathsf{s}\left[\substack{n-3\mathtt{M}\\n-7\mathtt{M}+1}\right] \oplus \mathcal{C}^{-1}\left(b_2\right) = \mathsf{s}\left[\substack{n-3\mathtt{M}\\n-7\mathtt{M}+1}\right] \oplus \mathcal{C}^{-1}\left(\mathsf{s}\left[\substack{n-\mathtt{M}\\n-3\mathtt{M}+1}\right] \oplus \mathcal{C}^{-1}\left(b_1\right)\right) \tag{5.4}$$

$$= \mathsf{s}\left[\substack{n-3\mathtt{M}\\n-7\mathtt{M}+1}\right] \oplus \mathcal{C}^{-1}\left(\mathsf{s}\left[\substack{n-\mathtt{M}\\n-3\mathtt{M}+1}\right] \oplus \mathcal{C}^{-1}\left(\mathsf{s}\left[\substack{n\\n-\mathtt{M}+1}\right]\right)\right) \tag{5.5}$$

$$\vdots \qquad\qquad \vdots \tag{5.6}$$

$$b_K \triangleq \mathsf{s}\left[\substack{\mathtt{M}2^{K-1}\\1}\right] \oplus \mathcal{C}^{-1}\left(b_{K-1}\right) = \mathsf{s}\left[\substack{\mathtt{M}2^{K-1}\\1}\right] \oplus \mathcal{C}^{-1}\left(\mathsf{s}\left[\substack{3\mathtt{M}2^{K-1}\\2^{K-1}\cdot\mathtt{M}+1}\right] \oplus \mathcal{C}^{-1}\left(b_{K-2}\right)\right) \tag{5.7}$$

$$= \mathsf{s}\left[\substack{\mathtt{M}2^{K-1}\\1}\right] \oplus \mathcal{C}^{-1}\left(\mathsf{s}\left[\substack{3\mathtt{M}2^{K-1}\\2^{K-1}\cdot\mathtt{M}+1}\right] \oplus \mathcal{C}^{-1}\left(\mathsf{s}\left[\substack{7\mathtt{M}2^{K-2}\\3\mathtt{M}2^{K-1}+1}\right] \oplus \ldots \oplus \mathcal{C}^{-1}\left(\mathsf{s}\left[\substack{n\\n-\mathtt{M}+1}\right]\right)\right)\right) \tag{5.8}$$

according to the general rule

$$b_i \triangleq \mathsf{s}\left[\substack{n-(2^{i-1}-1)\mathtt{M}\\n-(2^i-1)\mathtt{M}+1}\right] \oplus \mathcal{C}^{-1}\left(b_{i-1}\right). \tag{5.9}$$

The output of the encoder is the $\mathtt{M} \cdot 2^{K-1}$ bit sequence $b_K$ for the last block.

As we see from (5.7), $b_K$ is a description of the first block of source samples corrupted by the addition of $\mathcal{C}^{-1}\left(b_{K-1}\right)$. The decoder reconstructs the first $\mathtt{M} \cdot 2^{K-1}$ source samples via

$$\hat{\mathsf{s}}\left[\substack{\mathtt{M}2^{K-1}\\1}\right] \triangleq b_K = \mathsf{s}\left[\substack{\mathtt{M}2^{K-1}\\1}\right] \oplus \mathcal{C}^{-1}\left(b_{K-1}\right). \tag{5.10}$$

The distortion for this block is approximately $D_0$ since, by assumption, the decompresser $\mathcal{C}^{-1}\left(\cdot\right)$ maps its input to a sequence with a fraction $D_0$ ones. The error between the reconstruction and the true source obtained via feedforward is a description of future source samples shaped by the function $\mathcal{C}\left(\cdot\right)$. Thus, to reconstruct the next block, the decoder uses the feedforward, $\mathsf{s}\left[\substack{\mathtt{M}2^{K-1}\\1}\right]$, to produce

$$\hat{\mathsf{s}}\left[\substack{3\mathtt{M}2^{K-2}\\\mathtt{M}2^{K-1}+1}\right] \triangleq \mathcal{C}\left(\mathsf{s}\left[\substack{\mathtt{M}2^{K-1}\\1}\right] \oplus b_K\right) = b_{K-1} = \mathsf{s}\left[\substack{3\mathtt{M}2^{K-2}\\\mathtt{M}2^{K-1}+1}\right] \oplus \mathcal{C}^{-1}\left(b_{K-2}\right). \tag{5.11}$$

Once again the distortion is approximately $D_0$ since the decompresser maps its input to a sequence with $D_0$ ones.

The decoder proceeds in this manner and obtains a distortion of approximately $D_0$ for each block except the last which yields no distortion. The average distortion is therefore roughly $D_0$. Since $\mathtt{M}2^{K-1}$ bits are required to describe $b_K$ in encoding the $\mathtt{M}(2^K - 1)$ source samples, the average bit rate is $2^{K-1}/(2^K - 1) \approx 1/2$. Thus by taking advantage of the source feedforward, we can obtain a point on the rate distortion curve simply by using a low complexity lossless compression algorithm.

## 5.5 Finite Alphabet Sources & Arbitrary Distortion

In this section, we generalize the construction in Section 5.4 to arbitrary rates, source distributions and distortion measures. We require two components: a lossless compression/decompression algorithm and a shaping algorithm. Using these subsystems, we describe our feedforward source coding algorithm and present an analysis of its rate and distortion.

### 5.5.1 Feedforward Source Coding Subsystems

Our lossless compression and shaping algorithms must be efficient in some sense for the overall feedforward source coding algorithm to approach the rate-distortion function. Instead of delving into the details of how to build efficient compression and shaping algorithms, we define admissible systems to illustrate the required properties. We then describe how efficient subsystems can be combined.

#### Lossless Compression Subsystem

We define a $(\delta, \epsilon, m)$ admissible lossless compression system as follows. On input of $m$ samples from which are $\delta$-strongly typical[2] according to the distribution $p_{\hat{\mathsf{s}}}$, the compressor, denoted $\mathcal{C}_{\hat{\mathsf{s}}}\left(\cdot\right)$, returns $m \cdot H(\hat{\mathsf{s}}) + \epsilon$ bits. If the input is not $\delta$-strongly typical, the output is undefined. The corresponding decompresser, $\mathcal{C}_{\hat{\mathsf{s}}}^{-1}\left(\cdot\right)$ takes the resulting bits and reproduces the original input.

#### Shaping Subsystem

We define a $(\delta, \epsilon, m)$ admissible shaping system as follows. On input of a sequence of $m$ bits, and a semi-infinite sequence of samples, $\mathsf{s}\left[{}^{\infty}_{1}\right]$ which is $\delta$-strongly typical according to the distribution $p_{\mathsf{s}}$, the shaper $\mathcal{S}_{\hat{\mathsf{s}}|\mathsf{s}}\left(\cdot\right)$ returns a sequence of $m' = m \cdot [H(\hat{\mathsf{s}})/H(\hat{\mathsf{s}}|\mathsf{s})] + \epsilon$ samples, $\hat{\mathsf{s}}\left[{}^{m'}_{1}\right]$, such that $(\mathsf{s}\left[{}^{m'}_{1}\right], \hat{\mathsf{s}}\left[{}^{m'}_{1}\right])$ is $\delta$-strongly typical according to the distribution $p_{\hat{\mathsf{s}},\mathsf{s}}$. If the input is not $\delta$-strongly typical, the output is undefined. The corresponding deshaper takes the pair of sequences $\hat{\mathsf{s}}\left[{}^{m'}_{1}\right]$ and $\mathsf{s}\left[{}^{m'}_{1}\right]$ as input and returns the original sequence of $m$ bits.

The compression and shaping systems described previously are fixed-to-variable and variable-to-fixed systems respectively. Hence, for notational convenience we define the corresponding length functions $\mathcal{L}\left(\mathcal{C}_{\hat{\mathsf{s}}}\left(\cdot\right)\right)$ and $\mathcal{L}\left(\mathcal{S}_{\hat{\mathsf{s}}|\mathsf{s}}\left(\cdot\right)\right)$ as returning the length of their respective arguments.

#### Efficient Shaping and Compression Systems

We call a lossless compression system or a shaping system efficient if both $\delta$ and $\epsilon$ can be made arbitrarily small for $m$ large enough. Efficient lossless compression systems can be implemented in a variety of ways. For example, arithmetic coding is one well-known approach. Perhaps less well-known is that shaping systems can also be implemented via arithmetic coding [137]. Specifically, by using the decompresser for an arithmetic code as a shaper, we can map a sequence of bits into a sequence with an arbitrary distribution. The compressor for the arithmetic code takes the resulting sequence and returns the original bit sequence.

[2] A sequence is $\delta$-strongly typical if the empirical fraction of occurrences of each possible outcome differs by at most $\delta$ from the expected fraction of outcomes and no probability zero outcomes occur.

### 5.5.2 Feedforward Encoder and Decoder

Since the encoder for our feedforward lossy compression system is based on a variable-to-fixed shaper and a fixed-to-variable compressor, it is a variable-to-variable system. In practice, one could use buffering, padding, or other techniques to account for this when encoding a fixed length source or when required to produce a fixed length encoding. We do not address this issue further here. Instead, we assume that there is a nominal source block size parameter, $\mathtt{N}$, and buffering, padding, look-ahead, *etc.* is used to ensure that the system encodes $\mathtt{N}$ source samples (or possibly slightly more or less). Also, we assume that there is a minimum block size parameter, $\mathtt{M}$, which may be chosen to achieve an efficient shaping or lossless compression subsystem.

Once $\mathtt{N}$ and $\mathtt{M}$ are fixed, the feedforward encoder takes as input a stream of inputs $\mathsf{s}\left[{}^{\infty}_{1}\right]$ and encodes it as described in Table 5.1. The feedforward decoder takes as input the resulting bit string, $b$, and decodes it as described in Table 5.2. Section 5.4 describes an example of the encoding and decoding algorithm with a shaper (denoted $\mathcal{C}^{-1}(\cdot)$) mapping uniform bits to Bernoulli(0.11) bits. This example does not require a compressor because the $p_{\hat{\mathsf{s}}}$ distribution is incompressible.

**Table 5.1.** The Feedforward Encoder.

```
Initialize $T = 1$, $L = \mathtt{M}$, and reverse the input so that in the following $\mathsf{s}\left[{}^{n}_{1}\right] = \mathsf{s}\left[{}^{1}_{n}\right]$.
Take the block of source samples $\mathsf{s}\left[{}^{T+L}_{T}\right]$ and generate a "noisy version" $\hat{\mathsf{s}}\left[{}^{T+L}_{T}\right]$ (e.g., by generating each $\hat{\mathsf{s}}[i]$ from the corresponding $\mathsf{s}[i]$ according to $p_{\hat{\mathsf{s}}|\mathsf{s}}$).
while $L + T < \mathtt{N}$ do
    Compress $\hat{\mathsf{s}}\left[{}^{T+L}_{T}\right]$ to obtain the bit sequence $b = \mathcal{C}_{\hat{\mathsf{s}}}\left(\hat{\mathsf{s}}\left[{}^{T+L}_{T}\right]\right)$.
    $T \leftarrow T + L + 1$
    $L \leftarrow \mathcal{L}\left(\mathcal{S}_{\hat{\mathsf{s}}|\mathsf{s}}\left(b, \mathsf{s}\left[{}^{\infty}_{T}\right]\right)\right)$.
    $\hat{\mathsf{s}}\left[{}^{T+L}_{T}\right] \leftarrow \mathcal{S}_{\hat{\mathsf{s}}|\mathsf{s}}\left(b, \mathsf{s}\left[{}^{\infty}_{T}\right]\right)$
end while
return $\mathcal{C}_{\hat{\mathsf{s}}}\left(\hat{\mathsf{s}}\left[{}^{T+L}_{T}\right]\right)$
```

**Table 5.2.** The Feedforward Decoder.

```
Initialize $T$ to the length of the sequence encoded in $b$.
while $T > 1$ do
    $L \leftarrow \mathcal{L}\left(\mathcal{C}_{\hat{\mathsf{s}}}^{-1}(b)\right)$
    $T \leftarrow T - L + 1$
    $\hat{\mathsf{s}}\left[{}^{T+L}_{T}\right] \leftarrow \mathcal{C}_{\hat{\mathsf{s}}}^{-1}(b)$
    Get $\mathsf{s}\left[{}^{T+L}_{T}\right]$ via the feedforward information
    $b \leftarrow \mathcal{S}_{\hat{\mathsf{s}}|\mathsf{s}}^{-1}\left(\hat{\mathsf{s}}\left[{}^{T+L}_{T}\right], \mathsf{s}\left[{}^{T+L}_{T}\right]\right)$
end while
return the reversed version of $\hat{\mathsf{s}}\left[{}^{n}_{1}\right]$
```

## ■ 5.5.3 Rate-Distortion Analysis

**Theorem 19.** *By using efficient lossless compression and shaping subsystems, the distortion in encoding an i.i.d. sequence generated according to* $p_{\mathsf{s}}$ *can be made to approach* $E[d(\mathsf{s},\hat{\mathsf{s}},])$ *as closely as desired.*

*Proof.* First we note that by assumption, we can choose $\mathtt{M}$ large enough so that the probability of the source sequence being non-typical can be made negligible. For a typical source sequence, we can focus on how the encoder maps $\mathsf{s}\,[i]$ to $\hat{\mathsf{s}}\,[i]$ since the decoder simply maps a bit sequence to the $\hat{\mathsf{s}}\,[i]$ sequence chosen by the encoder. The encoder maps blocks of source samples, $\mathsf{s}\left[{}^{T+L}_{T}\right]$, to blocks of quantized samples, $\hat{\mathsf{s}}\left[{}^{T+L}_{T}\right]$, by using an admissible shaping algorithm. As described in Section 5.5.1, the shaper produces a $\delta$-strongly typical sequence. Thus the total expected distortion is at most

$$E[d(\mathsf{s},\hat{\mathsf{s}},]) + d_{\max}\cdot\delta + d_{\max}\cdot\Pr[\mathsf{s}\left[{}^{n}_{1}\right]\ \text{not typical}] \tag{5.12}$$

where the first two terms are the distortion for a typical sequence produced by the shaper and the last term is the contribution from a non-typical source sequence. □

**Theorem 20.** *By using efficient lossless compression and shaping subsystems, the rate in encoding an i.i.d. sequence generated according to* $p_{\mathsf{s}}$ *can be made to approach* $I(\mathsf{s};\hat{\mathsf{s}})$ *as closely as desired.*

*Proof.* Imagine that the parameter $\mathtt{N}$ is chosen so that $K$ passes of the loop in the encoding algorithm are executed. Also, let $L_j$ denote the value of $L$ in line 3 of the encoder in the $j$th pass. We know $L_1=\mathtt{M}$ by construction. By definition of an admissible shaping system in Section 5.5.1 and line 6 of the encoder we have that $L_{j+1}\geq L_j\cdot[H(\hat{\mathsf{s}})/H(\hat{\mathsf{s}}|\mathsf{s})]$. Using this relation and assuming that each block of length $L_j$ is typical, we can compute the total number of samples encoded via

$$n=\sum_{j=1}^{K}L_j\geq\sum_{j=0}^{K-1}\mathtt{M}\cdot\left[\frac{H(\hat{\mathsf{s}})}{H(\hat{\mathsf{s}}|\mathsf{s})}\right]^j=\mathtt{M}\cdot\frac{[H(\hat{\mathsf{s}})/H(\hat{\mathsf{s}}|\mathsf{s})]^K-1}{H(\hat{\mathsf{s}})/H(\hat{\mathsf{s}}|\mathsf{s})-1}. \tag{5.13}$$

The bit rate required to encode these samples is

$$R=L_K\cdot H(\hat{\mathsf{s}})+\epsilon\leq\mathtt{M}\cdot H(\hat{\mathsf{s}})\cdot[H(\hat{\mathsf{s}})/H(\hat{\mathsf{s}}|\mathsf{s})]^{K-1}+\epsilon\cdot K\cdot[H(\hat{\mathsf{s}})/H(\hat{\mathsf{s}}|\mathsf{s})]^K. \tag{5.14}$$

This follows by the assumption that the admissible lossless compression system in Section 5.5.1 requires $m\cdot H(\hat{\mathsf{s}})+\epsilon$ bits to encode a block of $m$ typical samples.

Therefore the number of bits per sample when the source blocks are typical is obtained by dividing (5.14) by (5.13) to obtain

$$R/n\leq\left\{\mathtt{M}\cdot H(\hat{\mathsf{s}})\cdot\left[\frac{H(\hat{\mathsf{s}})}{H(\hat{\mathsf{s}}|\mathsf{s})}\right]^{K-1}+\epsilon\cdot K\cdot\left[\frac{H(\hat{\mathsf{s}})}{H(\hat{\mathsf{s}}|\mathsf{s})}\right]^K\right\}\Big/\left\{\mathtt{M}\cdot\frac{[H(\hat{\mathsf{s}})/H(\hat{\mathsf{s}}|\mathsf{s})]^K-1}{H(\hat{\mathsf{s}})/H(\hat{\mathsf{s}}|\mathsf{s})-1}\right\} \tag{5.15}$$

$$=\left\{H(\hat{\mathsf{s}})\left[1-\frac{H(\hat{\mathsf{s}}|\mathsf{s})}{H(\hat{\mathsf{s}})}\right]+\frac{\epsilon K}{\mathtt{M}}\left[\frac{H(\hat{\mathsf{s}})}{H(\hat{\mathsf{s}}|\mathsf{s})}-1\right]\right\}\Big/\left\{1-\left[\frac{H(\hat{\mathsf{s}})}{H(\hat{\mathsf{s}}|\mathsf{s})}\right]^{-K}\right\} \tag{5.16}$$

$$=I(\hat{\mathsf{s}};\mathsf{s})\cdot\left\{1+\frac{\epsilon K}{\mathtt{M}H(\hat{\mathsf{s}}|\mathsf{s})}\right\}\Big/\left\{1-\left[\frac{H(\hat{\mathsf{s}})}{H(\hat{\mathsf{s}}|\mathsf{s})}\right]^{-K}\right\}. \tag{5.17}$$

An extra term must also be added to account for the possibility that the source is atypical. By assumption we can choose $\mathtt{N}$ so that $K$ is large enough to make the second term in braces negligible, and then we can choose $\mathtt{M}$ so that the probability of any source block being typical is negligible. Also, by making $\mathtt{M}$ large enough we can make the first term in curly braces negligible. Thus the bit rate can be made as close to $I(\hat{s}; s)$ as desired.

□

Combining the previous theorems indicates that we can approach the feedforward rate-distortion function with only the complexity required for lossless compression and shaping systems.

**Corollary 5.** *When linear complexity admissible lossless compression and shaping systems are used, the resulting feedforward rate-distortion function can be approached arbitrarily closely with linear complexity.*

In particular, we can use the lossless compression and shaping systems described in [137] which are based on arithmetic coding and the dual of arithmetic coding respectively.

## ■ 5.6 Concluding Remarks

In this chapter we describe a lossy compression algorithm to encode a finite-alphabet source in the presence of feedforward information. In particular, we show that although memoryless feedforward does not change the rate-distortion function, it allows us to construct a low complexity lossy compression system which approaches the rate-distortion function. In practice, the particular scheme described here may require modifications and other methods of using feedforward information or similar knowledge may be more appropriate. Our main goal therefore is not necessarily to advocate a particular scheme but to show that when compression, observation, and control interact, additional resources such as feedforward may provide advantages not available in the classic compression framework. One interesting possibility for future work includes studying the general problem in Section 5.2 when the fixed lag side information, $w$, is not exactly the same as the source. Similarly, investigating the effects of memory in the source and different values for the delay, $\Delta$, would also be valuable.

# Part II

# Delay Constraints

Chapter 6

# System Model

In this section we outline the system model we plan to study in this part of the thesis. Our main goal is to introduce a general model motivated by delay sensitive applications, discuss the key features of this model and note its relationship to previous work. Later chapters further specialize the model when necessary.

As demonstrated by Shannon [162], arbitrarily low error probability can be achieved in transmitting messages across memoryless, noisy channels provided that the transmission rate is below the channel capacity and long channel codes are used. Similarly, a source can be encoded with minimum distortion provided long source codes are used in quantizing the source. However, when the channel suffers occasional impairments due to signal fading, shadowing, mobility, congestion, interference, etc., and delay constraints prohibit averaging over these impairments, Shannon capacity may no longer be a relevant measure.

Specifically, achieving the (ergodic) Shannon capacity of a channel requires coding over a long enough block that the transmission encounters each channel state in proportion to its probability. When latency constraints limit the length of the transmission, arbitrarily low error probability may be unattainable at rates approaching the Shannon capacity. Similarly, when the source statistics vary due to motion, sudden activity, or other correlations, achieving optimal distortion may be impossible due to latency constraints. In the sequel, we present these issues in the channel coding framework with the understanding that our models can be translated to the source coding problem in a straight-forward manner.

### ■ 6.0.1 Notation

In this part of the thesis we continue our notational conventions from the previous part. Vectors and sequences are denoted in bold (*e.g.*, $\mathbf{x}$) with the $i$th element denoted as $x[i]$. Random variables are denoted using the sans serif font (*e.g.*, $\mathsf{x}$) while random vectors and sequences are denoted with bold sans serif (*e.g.*, $\boldsymbol{\mathsf{x}}$). Subsequences are denoted by stacking (*e.g.*, $x\left[{}^{j}_{i}\right]$ denotes the subsequence $x[i]$, $x[i+1]$, …, $x[j]$).

## ■ 6.1 Channel Models

A general (memoryless) channel model which captures these effects can be constructed by letting the channel output vector, $\boldsymbol{\mathsf{y}}$, depend on the channel input vector, $\boldsymbol{\mathsf{x}}$, with a different probability law for each sample

$$p_{\boldsymbol{\mathsf{y}}|\boldsymbol{\mathsf{x}}}(\mathbf{y}|\mathbf{x}) = \prod_i p_{\mathsf{y}|\mathsf{x};\boldsymbol{\theta}}(y[i]\,|x[i]\,;\theta[i]). \tag{6.1}$$

where $\boldsymbol{\theta}[0]$, $\boldsymbol{\theta}[1]$, … denote channel states (*e.g.*, modeling the level of fading or amount of interference).

To model packet based systems as well as systems where the channel state changes relatively slowly, researchers often use piece-wise constant or block-interference channel models [126] [23] [93] [141]. In such channels the inputs and outputs are grouped into blocks $\boldsymbol{\mathsf{x}}[0]$, $\boldsymbol{\mathsf{x}}[1]$, …, $\boldsymbol{\mathsf{x}}[\mathtt{T}]$ and $\boldsymbol{\mathsf{y}}[0]$, $\boldsymbol{\mathsf{y}}[1]$, …, $\boldsymbol{\mathsf{y}}[\mathtt{T}]$ where each block consists of $n_c$ samples: $\boldsymbol{\mathsf{x}}[i] = \mathsf{x}_0[i], \mathsf{x}_1[i], \ldots, \mathsf{x}_{n_c-1}[i]$ and $\boldsymbol{\mathsf{y}}[i] = \mathsf{y}_0[i], \mathsf{y}_1[i], \ldots, \mathsf{y}_{n_c-1}[i]$. The channel law is constant for each block:

$$\prod_{i=0}^{\mathtt{T}} p_{\boldsymbol{\mathsf{y}}[i]|\boldsymbol{\mathsf{x}}[i];\boldsymbol{\theta}[i]}(\mathbf{y}[i]\,|\mathbf{x}[i]\,;\theta[i]) = \prod_{i=0}^{\mathtt{T}}\prod_{j=1}^{n_c} p_{\mathsf{y}|\mathsf{x};\boldsymbol{\theta}}(y_j[i]|x_j[i];\theta[i])$$

where the block size, $n_c$, typically depends on problem details such as the coherence time in wireless communication or the Maximum Transmit Unit (MTU) in wired networks. When coding is allowed only over a single block (*i.e.*, $\mathtt{T} = 0$), this (non-ergodic) channel is sometimes referred to as the compound channel [50] or a class of channels [25], [187].

Ideally, $\mathtt{T}$ should be large enough that the receiver can average over the channel variations and obtain arbitrarily low probability of error provided that the transmission rate is below the ergodic capacity. However, due to delay constraints, $\mathtt{T}$ is often constrained to be finite and reliably transmitting rates approaching the ergodic capacity is not possible. In this case, previous researchers have characterized performance using measures such as outage capacity [141] [23] or delay-limited capacity [81].

While such work highlights some key issues in dealing with time-varying channel uncertainty, in order to focus on real-time applications where delay and causality are important constraints, we consider a sequential model which requires block Markov coding [45]. Specifically, we consider the model where the transmitter has a sequence of messages, $\mathbf{s}[i]$, to send over the channel. At each time step, the transmitter chooses the channel input $\mathbf{x}[i]$ as a function of $\mathbf{s}[i]$, $\mathbf{s}[i-1]$, ... and the receiver obtains the corresponding channel output $\mathbf{y}[i]$. The receiver's goal is to reliably estimate $\mathbf{s}[i]$ within a given delay constraint (to be defined shortly).

There are two key differences between the sequential model and previous models. First, instead of sending a fixed length message, $\mathbf{s}$, known before transmission begins, the encoder must send a stream of messages, $\mathbf{s}[i]$, revealed causally. Second, instead of only measuring whether the entire message (or even a piece of the message) can be decoded *eventually*, we are interested in the decoding delay. As we shall argue, the sequential model is suited to delay sensitive applications such as video or audio streaming, two-way communications, remote control, and many others. Furthermore, we shall demonstrate that it leads to a rich set of new algorithms for real-time communication.

## ■ 6.2 Notions Of Delay

Many formulations of communication problems with delay exist. Before we choose a model and explore the resulting conclusions, it is useful to consider what issues we are trying to understand with a given definition of delay. For example, error exponents are one common notion of delay (*i.e.*, a block notion). The analysis of error exponents essentially asks "How large a block is required to achieve the performance promised by random coding arguments and the law of large numbers?" The theory of error exponents answering this question is useful in deciding how large a block (and correspondingly how much latency) to use in a block code. But we are interested in a different question: "How do we design codes which can be quickly decoded despite dynamic channel fluctuations?". Therefore, after discussing various notions of delay we introduce the idea of "packet delay" we use in the rest of the thesis.

### ■ 6.2.1 Block-Delay

Consider transmitting a message which consists of one long file. For such messages, a natural performance measure is the block-delay: the length of time between when the encoder starts transmitting and when the receiver decodes the entire message. Generally, block-delay is an appropriate performance measure for monolithic messages where no piece of the message is useful without the whole. Block-delay seems to be one of the earliest measure of delay and is connected to the traditional analysis of error exponents (for standard channels) as well as outage-capacity or delay-limited capacity (for compound channels). Recently, codes designed to minimize the block-delay in transmitting large files have been reported [115,31,163,59,53].

However, certain information sources possess a natural ordering of distinct segments (*e.g.*, video, audio, sequential sensor readings, or statements in two-way conversations). Decoding earlier portions of a message of this kind before the entire message is received is not only valuable but sometimes essential. For such systems, other measures of delay are desirable.

### ■ 6.2.2 Symbol-Delay

Perhaps the most basic way to measure decoding delay in a sequential model is via the symbol-delay *i.e.*, the number of time units between when $s[i]$ is sent and it is decoded (neglecting any propagation delay). Essentially, this corresponds to considering packets of length $n_c = 1$. For example, if a convolutional encoder with constraint length $L$ is used with a Viterbi decoder, then it is well-known that a decoding delay of $5L$ causes a negligible loss in performance [151]. The symbol-delay (or in this case the bit-delay) for a system designed to make decisions according to this criteria would be simply $5L$ as illustrated in Fig. 6-1 for the case of $L = 4$.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Source Sequence: | $s[0]$ | $s[1]$ | ... | $s[20]$ | $s[21]$ | $s[22]$ | $s[23]$ | $s[24]$ | $s[25]$ |
| Encoded Sequence: | $x[0]$ | $x[1]$ | ... | $x[20]$ | $x[21]$ | $x[22]$ | $x[23]$ | $x[24]$ | $x[25]$ |
| Channel Output: | $y[0]$ | $y[1]$ | ... | $y[20]$ | $y[21]$ | $y[22]$ | $y[23]$ | $y[24]$ | $y[25]$ |
| Decoder Output: | ... | ... | ... | $\hat{s}[0]$ | $\hat{s}[1]$ | $\hat{s}[2]$ | $\hat{s}[3]$ | $\hat{s}[4]$ | $\hat{s}[5]$ |

**Figure 6-1.** Illustration of a system with a bit-delay of 20.

However, if long delays are undesirable, some symbols may be decoded earlier and some later resulting in a different symbol-delay for each message element. In such cases, statistical characteristics of the symbol-delay (*e.g.*, moments of the symbol-delay, the symbol-delay histogram, or the decay rate of the histogram) may be appropriate.

### ■ 6.2.3 Packet-Delay

Symbol-delay may seem like the most complete delay characterization (though potentially intractable to analyze). For example, symbol-delay subsumes block-delay in the sense that if all the symbols are presented to the encoder simultaneously, then the maximum symbol-delay is exactly the block-delay. However, the concept of symbol-delay fails to capture a key facet of many practical systems: packetization.

Consider a music or video streaming system or a video-conferencing system. In such systems the encoder receives a sequence of audio or video samples to quantize from the microphone or camera. Usually, however, these source samples are not quantized as they are received but blocked into a packet and encoded together in order to apply powerful source coding techniques such as vector quantization, LPC coding, wavelet transforms, *etc.* Furthermore, at the physical layer, information is usually collected into packets due to the benefits of packet switched communications.

Thus measuring the time interval between when a certain audio or video *sample* entered the encoder and was reconstructed at the receiver is not especially meaningful. Instead, it is more useful to consider the time interval between when a source *packet* entered the encoder and when it was reconstructed by the decoder. Consequently, for packet based systems we introduce the following notion of packet-delay.

Let the encoder input be a sequence of message packets $\mathbf{s}[i]$ where each message packet is composed of $n_s$ samples $s_0[i]$, $s_1[i]$, $\ldots$, $s_{n_s-1}[i]$. The encoder transforms the input packet stream into a sequence of channel input packets of length $n_c$, $\mathbf{x}[i] = x_0[i], x_1[i], \ldots, x_{n_c-1}[i]$, which are transmitted to the receiver. A decoder takes the resulting channel output packets, $\mathbf{y}[i]$, and produces an estimate of the original message $\hat{\mathbf{s}}[i]$. The packet-delay is the number of time units between when $\mathbf{s}[i]$ is sent and $\hat{\mathbf{s}}[i]$ is decoded. Essentially, one can think of coding in the packet-delay setting as using a type of convolutional code where the basic unit is a "super-symbol" instead of a bit and each super-symbol corresponds to a single packet.

In the sequel, we will focus on packet-delay since it most closely models practical systems of interest. For many voice, video and audio applications and many physical layer designs, packet sizes are reasonably large. Consequently, we study limiting performance as the packet size grows to infinity (*i.e.*, as $n_s \to \infty$ or $n_c \to \infty$). At first glance, it may seem strange that when $n_s \to \infty$ or $n_c \to \infty$ the symbol-delay grows to infinity, but the packet-delay remains finite and well-defined. We emphasize that in practice, limiting performance can be achieved with packet sizes on the order of hundreds of bits (*e.g.*, see Chapter 8 and Section 9.4). Hence the packet-delay model illustrates system performance when packets are reasonably large, but coding can be performed over a limited, finite number of channel states.

Chapter 7

# Streaming Codes For Bursty Channels

In a number of settings, channel fluctuations occur in bursts. Some examples include packet losses due to congestion in the Internet [143] [27] [191] [29] [13], as well as multipath fading, shadowing, interference, etc. In wired packet networks, burstiness can be caused by switch or router queues overflowing and dropping many consecutive packets, as well as by protocol dynamics, and cross-traffic. In wireless networks fading, shadowing, and interference typically have time scales that are determined by the receiver or transmitter mobility, carrier frequency, or terrain. Perhaps due to the many possible causes of burstiness, there is no canonical bursty channel model. Specifically, although bursty channel models have been studied since the early days of information theory [73], there is little agreement on which channel models are the most detailed, accurate and analytically tractable.

Consequently, we focus mainly on an adversarial channel model since it requires fewer parameters or assumptions and represents a kind of conservative worst-case design. Furthermore, since we are mainly interested in the interplay between channel dynamics and efficient codes, our idealized channel model highlights these issues most clearly. The resulting code constructions illustrate what seems to be a new idea in coding. Of course, these codes are intended to be used in real channels and the performance of our codes on more detailed channel models is an important topic which we defer to other work [119, 121].

We begin by studying block burst models which are similar to traditional block-fading models [141] [93] used in wireless communication, and block erasure channels used in packet networks. In such models, the channel characteristics are constant for each block of inputs (which we call a packet), but occasionally the channel suffers a burst of noise, interference or other such effects which last for a few blocks or packets. To explore the benefits of coding over both time and space, we consider a model with $\mathtt{M}$ different transmission paths. For simplicity, we consider a channel with two possible states where at time $i$, the block transmitted over path $j$ may be received either over the good channel state or the bad channel state.

Our goal is to develop codes which can recover from a burst of $\mathtt{B}$ consecutive occurrences of the bad channel state with the minimum possible delay. We first illustrate the problem with a brief Gaussian example in Section 7.1. Then, after more precisely defining our channel model in Section 7.2, we introduce and compute the burst-delay capacity in Section 7.3. Section 7.4 shows that no rates above the burst delay capacity can be reliably communicated under a decoding delay constraint of $\mathtt{T}$ packets while Section 7.5 provides a random coding construction achieving all rates below the burst-delay capacity. Finally, Section 7.6 presents some detailed examples of the random coding arguments, Section 7.7 discusses more practical coding structures, and Section 7.8 provides some concluding remarks.

## 7.1 A Gaussian Example

Consider a block fading, complex additive white Gaussian noise channel used to send an information stream. Specifically, imagine that at each time $i$, a data source produces a packet of information (*e.g.*, a packet of speech), $\mathbf{s}[i]$, which the transmitter maps into an encoded block $\mathbf{x}[i]$ to send through the channel. [1] Ideally, the receiver decodes each block as soon as it is received. Occasionally, however, channel fluctuations such as fading may make it impossible to decode block $i$ when it is received. Thus, the goal is to using coding so that, in such cases, the data from block $i$ can be decoded from later blocks. If the packet

[1] Note that each channel input block can only depend on current and past source blocks and so the channel coding must be causal in some sense.

produced by the data source at time $i$ is decoded when block $j$ is received, we say that a decoding delay of $j - i$ is incurred.

For simplicity, consider a scenario where nominally each block is received with a signal-to-noise ratio (SNR) of 10 dB and the data for that block can be reliably decoded with negligible error probability provided the block is large enough.[2] Furthermore, imagine that occasionally a burst of $\mathtt{B}$ blocks are received with a lower SNR of 0 dB due to fading. What is the decoding delay required to recover the information transmitted in the faded blocks?

The unfaded state can support a rate of $\log 11$ bits per sample while the faded state can only support a rate of $\log 2$ bits per sample. Thus if data is sent at the latter rate, it can always be decoded with no delay regardless of fading. By contrast, if data is sent at the former rate, then the faded blocks can never be decoded resulting in an infinite delay for the corresponding source packets. Intuitively, as the transmission rate is increased from $\log 2$ to $\log 11$ the decoding delay should increase from 0 to $\infty$, but how does the optimal decoding delay vary as a function of the rate and what is the appropriate coding scheme?

For example, one possible option is to repeat the information in block $i$ in block $i + \mathtt{B}$. This would result in a decoding delay of $\mathtt{B}$ provided that packet $i$ contains at most $(1/2)\log 11$ bits per sample of new information plus the $(1/2)\log 11$ bits per sample of information repeated from time $i - \mathtt{B}$. Evidently, repetition coding achieves a decoding delay of $\mathtt{B}$ while transmitting at a rate of $(1/2)\log 11$ bits per sample. Another approach would be to encode the information stream with a convolutional code with maximum free distance or with an optimal distance profile. Although the decoding delay and rate are harder to analyze for convolutional codes than for repetition codes, all of these coding schemes are in fact suboptimal.

Specifically, there exists a coding scheme which achieves a decoding delay of $\mathtt{T}$ with a rate of

$$\frac{\mathtt{T} \cdot \log 11 + \mathtt{B} \log 2}{\mathtt{T} + \mathtt{B}} \tag{7.1}$$

for all $\mathtt{T} \geq \mathtt{B}$. The causality requirement implies that for $\mathtt{T} < \mathtt{B}$, the best strategy is to code for the worst case and send only $\log 2$ bits per sample in every block. Furthermore, it is possible to show that no coding scheme can achieve a lower decoding delay at the same rate.

The point of this example is to illustrate the type of delay issues we plan to study, to show that the decoding delay vs. transmission rate trade-off is interesting and non-trivial, and to show that the most obvious coding schemes are suboptimal. In the sequel, we examine more general channels and consider space or path diversity techniques in addition to coding over time.

## ■ 7.2 Problem Model

We consider the problem model in Fig. 7-1 which allows coding over both time and space in order to study the decoding delay required to correct bursts. Specifically, at each time $i$, the transmitter chooses $\mathtt{M}$ channel input blocks $\mathbf{x}[i,p]$ with $p$ ranging from 0 to $\mathtt{M}-1$. The different paths could correspond to different frequency bands or antennas, in wireless communication or to different routes in wired networks. Each $\mathbf{x}[i,p]$ consists of $n_c$ components $x_0[i,p]$, $x_1[i,p]$, $\ldots$, $x_{n_c-1}[i,p]$ all taking values in the finite alphabet $\mathcal{X}$. The channel has a good

[2] In practice, the size of each block will depend on the coherence time of the channel, but for now we suppress this detail and assume that the blocks are large enough that the bursts are the dominant cause of error.

state, $\theta_{\mathcal{G}}$, where the channel outputs for each path (taking values in the alphabet $\mathcal{Y}$) are determined according to the (memoryless) "good channel law"

$$p_{\mathbf{y}[i,p]|\mathbf{x}[i,p]}(\mathbf{y}[i,p]|\mathbf{x}[i,p]) = \prod_{j=1}^{n_c} p_{y|x}(y_j[i,p]|x_j[i,p];\theta_{\mathcal{G}}). \tag{7.2}$$

and a bad state, $\theta_{\mathcal{B}}$, where the channel outputs for path $p$ are determined according to the (memoryless) "bad channel law"

$$p_{\mathbf{y}[i,p]|\mathbf{x}[i,p]}(\mathbf{y}[i,p]|\mathbf{x}[i,p]) = \prod_{j=1}^{n_c} p_{y|x}(y_j[i,p]|x_j[i,p];\theta_{\mathcal{B}}). \tag{7.3}$$

We assume that the receiver knows which channel law is in effect, but the transmitter does not. The packet size, $n_c$, is discussed in more detail at the end of Section 7.2.1.

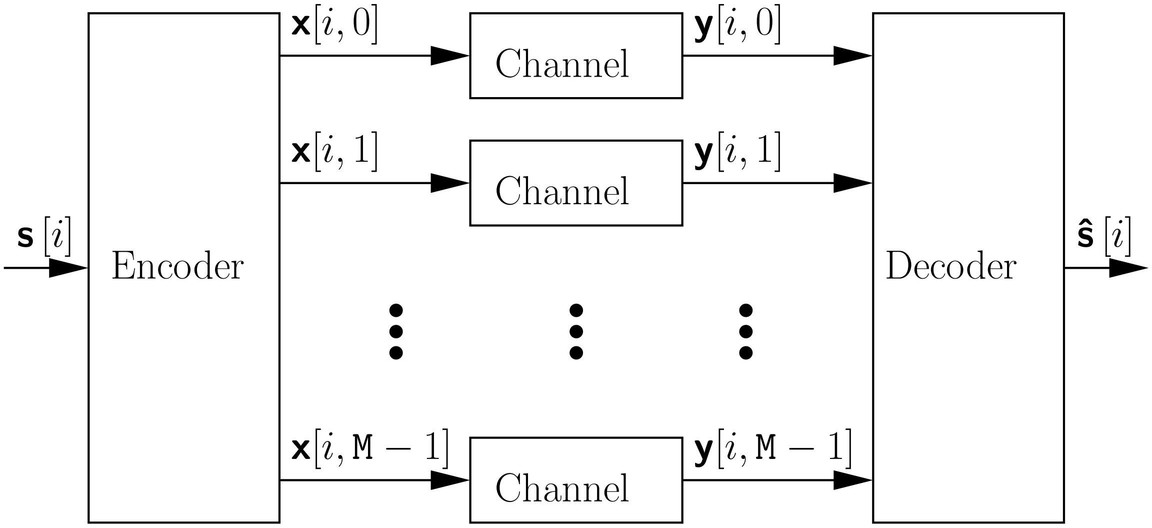


**Figure 7-1.** The low delay burst correction problem model. At each time $i$, the encoder observes the source packet $\mathbf{s}[i]$ and produces $\mathtt{M}$ channel input packets for the $\mathtt{M}$ possible paths which could correspond to different frequency bands, antennas, in a wireless network or different routes in a wired network. A burst of length $\mathtt{B}$ may occur on any single path and the decoder must be able to correct all such bursts with delay $\mathtt{T}$.

We define an $\mathtt{M}$ path packet encoder, $f(\cdot)$, as a causal mapping from a source symbol stream, $\mathbf{s}[i]$, consisting of $n_s$ components, $\mathsf{s}_j[i]$ each over the alphabet $\mathcal{S}$, to a sequence of $\mathtt{M}$ symbol streams $\mathbf{x}[i,0]$, $\mathbf{x}[i,1]$, ..., $\mathbf{x}[i,\mathtt{M}-1]$,

$$f(\cdot) : \mathcal{S}_{-\infty}^{n_s \cdot i} \mapsto \mathcal{X}^{n_c}. \tag{7.4}$$

We define a packet decoder operating with delay $\mathtt{T}$, $g_{\mathtt{T}}(\cdot)$, as a mapping from the received signal stream up to time $i+\mathtt{T}$ to an estimate of the source at time $i$:

$$g_{\mathtt{T}}(\cdot) : \mathcal{Y}_{-\infty}^{n_c \cdot (i+\mathtt{T})} \mapsto \mathcal{S}^{n_s}. \tag{7.5}$$

In contrast to the standard communication model where a single, finite-length message is communicated using one block of channel uses, our model considers an information stream where new information is continuously generated and sent. Consequently, to focus on the delay required to correct occasional channel fluctuations, we need a slightly different notion of probability of error. Therefore, we define the maximal, delay $\mathtt{T}$, block-based probability of error, $P_e(\boldsymbol{\theta}, \mathtt{T})$, for a given sequence of channel states, $\boldsymbol{\theta}$ as

$$P_e(\boldsymbol{\theta}, \mathtt{T}) \stackrel{\Delta}{=} \max_i \max_{\mathbf{s}[i]} \Pr\left\{ g_{\mathtt{T}}(\mathbf{y}\left[i+\mathtt{T}\right], \mathbf{y}\left[i+\mathtt{T}-1\right], \ldots|\boldsymbol{\theta}) \neq \mathbf{s}\left[i\right] \right\}. \tag{7.6}$$

The inner max is the familiar maximum of the probability of error taken over all messages while the outer max ensures that the probability of error is low for all times. For the usual case where the source consists of bits, (*i.e.*, $\mathcal{S} = \{0, 1\}$), the rate of a system is defined as the number of source bits per channel sample:

$$R \stackrel{\Delta}{=} \frac{n_s}{n_c} \tag{7.7}$$

### ■ 7.2.1 Defining Burst-Delay Capacity

To compare various communication systems and study the trade-off between burst length, decoding delay, and rate, we need to describe how the channel state sequence, $\boldsymbol{\theta}$, is chosen and define a performance metric. One approach would be to specify a probabilistic model for $\boldsymbol{\theta}$ and investigate various notions of capacity such as outage-capacity [141] [23], delay-limited capacity [81], or ergodic Shannon capacity. One drawback to this approach is that determining an accurate probabilistic model for the channel state sequences is often difficult in practice. Also, since it is generally impossible to guarantee any finite rate can be reliably communicated for a probabilistic state sequence, all of our analysis would depend strongly on the error probability. Finally, in our opinion, the difficulty in analyzing probabilistic models seems to obscure some essential features of the relationship between channel dynamics and code design.

Consequently we consider an adversarial channel model where the channel is nominally in the good state. At any time, however, the channel may enter the bad state for $\mathtt{B}$ consecutive blocks. Formally, we define the $\mathtt{B}$ burst ensemble, $\Theta_{\mathtt{B}}$, as all state sequences with exactly one burst of length $\mathtt{B}$:

$$\Theta_{\mathtt{B}} = \left\{ \boldsymbol{\theta} | \theta\left[i\right] = \theta_{\mathcal{B}}, i^* \leq i \leq i^* + \mathtt{B} - 1, \text{ and } \theta\left[i\right] = \theta_{\mathcal{G}} \text{ otherwise} \right\}. \tag{7.8}$$

In practice, the channel may experience multiple bursts of the bad channel state, and provided that such bursts are far enough apart, the results for the single burst ensemble, $\Theta_{\mathtt{B}}$, will also apply to multiple burst channels as well. An important advantage of the adversarial model is that it is in some sense a weaker assumption about the channel since we do not presume to know exactly how often bursts occur, but only require knowledge of a characteristic length. A second advantage of this model is that it admits the following simple but powerful notion of capacity.

We define the *operational* burst-delay capacity $C_{\mathtt{B,T}}$ as the maximum rate such that there exist encoders and decoders where the maximal, delay $\mathtt{T}$, block-based probability of error in (7.6) for any channel state sequence in $\Theta_{\mathtt{B}}$ goes to zero asymptotically with $n_c$:

$$C_{\mathtt{B,T}} \stackrel{\Delta}{=} \sup R \text{ s.t. } \exists f(\cdot), g_{\mathtt{T}}(\cdot) \text{ with } \lim_{n_c \to \infty} P_e(\boldsymbol{\theta}, \mathtt{T}) = 0 \ \forall \boldsymbol{\theta} \in \Theta_{\mathtt{B}}. \tag{7.9}$$

It may seem strange that (7.9) chooses a scaling where the block size, $n_c$, goes to infinity while considering a finite delay. In practice, we are interested in systems with a fixed, finite, but reasonably large, packet size $n_c$. For example, in wireless or wired networks $n_c$ may be determined by the coherence time or the Maximum Transmit Unit (MTU), respectively. Our goal is to focus on the channel dynamics and therefore the lowest meaningful time-scale for the delay is the number of channel blocks or packets required to recover from a burst. Since finite $n_c$ effects only serve to complicate the analysis and do not affect the fundamental trade-offs when packets are "large enough", we focus on asymptotic results. By using the theory of error exponents, it is straightforward to show that our asymptotic analysis becomes valid for quite modest packet sizes as shown in Chapter 8.

## ■ 7.3 Computing Burst-Delay Capacity

There are essentially two issues in computing the burst delay capacity: choosing the proper input distribution for the channel and choosing a coding scheme with the right dynamics. We are most interested in the latter and so we focus on the case where the two channel laws $p_{\mathsf{y}|\mathsf{x}}(\cdot|\cdot;\theta_{\mathcal{B}})$ and $p_{\mathsf{y}|\mathsf{x}}(\cdot|\cdot;\theta_{\mathcal{G}})$ are such that a single input distribution maximizes the mutual information for both. We refer to such a pair of channel laws as input compatible. For example, Gaussian channels with different signal-to-noise ratios are input compatible as are binary erasure channels and binary symmetric channels. In contrast, a pair of channels with additive Gaussian noise and additive Laplacian noise are not input compatible. In any case, we define the *informational* burst-delay capacity as

$$C^{(I)}_{\mathtt{B},\mathtt{T}} \triangleq \max_{p_{\mathsf{x}}(x)} \begin{cases} \mathtt{M} \cdot I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - \frac{\mathtt{B}}{\mathtt{B}+\mathtt{T}} \cdot [I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})], & \mathtt{T} \geq \mathtt{B} \\ (\mathtt{M}-1) \cdot I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}}) + I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})], & \mathtt{T} < \mathtt{B} \end{cases} \tag{7.10}$$

from which we obtain the following theorem.

**Theorem 21.** *For a pair of input compatible channel laws, the informational and operational burst-delay capacities are equal.*

*Proof.* We prove the two parts of Theorem 21 separately. Specifically, Theorem 22 in Section 7.4 shows no better rates are achievable while Theorem 23 at the end of Section 7.5 shows that all rates up to $C_{\mathtt{B},\mathtt{T}}$ are achievable. □

We conjecture that the theorem holds even for non input compatible channels. Specifically, the rate in (7.10) is actually achievable for any pair of channels, but we do not yet have a proof that a higher rate is impossible for non input compatible channels.

Before moving on to proofs, we examine some of the implications of Theorem 21. Evidently, for delays longer than the burst length, the rate loss imposed by the delay constraint is exactly

$$\frac{\mathtt{B}}{\mathtt{B}+\mathtt{T}} \cdot [I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})]\,. \tag{7.11}$$

For example, consider a single path, (*i.e.*, $\mathtt{M}=1$), burst erasure channel (*i.e.*, $I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})=0$) where the good channel state corresponds to perfect reception (*i.e.*, $I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}})=1$). Then the rate loss for decoding all bursts of $\mathtt{B}$ erased packets with a delay of $\mathtt{T}$ packets is $\mathtt{B}/(\mathtt{T}+\mathtt{B})$ for $\mathtt{T} \geq \mathtt{B}$ as derived previously in [119]. For example, if we insist that $\mathtt{T}=\mathtt{B}$, then the burst-delay capacity is $1/2$. This rate can be achieved simply by placing the source data at time $i$, $\mathbf{s}[i]$, in the first half of the $i$th channel input, $\mathbf{x}[i]$, and repeating it again in the second half of $\mathbf{x}[i+\mathtt{B}]$.

Alternatively, we can also consider a single path fading channel where the good channel state corresponds to a complex, additive white Gaussian noise (AWGN) channel with signal-to-noise ratio $\mathrm{SNR}_1$ and the bad channel state corresponds to an AWGN channel with signal-to-noise ratio $\mathrm{SNR}_2$. Once again if we require $\mathtt{T} = \mathtt{B}$, then the burst-delay capacity is

$$\frac{1}{2}\left[\log(1+\mathrm{SNR}_1)+\log(1+\mathrm{SNR}_2)\right] \tag{7.12}$$

and the rate loss for meeting the delay constraint is

$$\frac{1}{2}\left(\log(1+\mathrm{SNR}_1)-\log(1+\mathrm{SNR}_2)\right). \tag{7.13}$$

As with the erasure channel, this rate can be achieved by using half the bandwidth in $\mathbf{x}[i]$ for $\mathbf{s}[i]$ and the other half for $\mathbf{s}[i-\mathtt{B}]$. For the Gaussian case, however, a more complicated decoding process similar to maximal ratio combining is required. When a burst occurs at time $i-\mathtt{B}$, the receiver waits until $\mathbf{y}[i]$ is received. Then the receiver jointly decodes $\mathbf{s}[i]$ and $\mathbf{s}[i-\mathtt{B}]$ from $\mathbf{y}[i]$ and $\mathbf{y}[i-\mathtt{B}]$. The latter provides $\log(1+\mathrm{SNR}_1)$ bits of mutual information while the former provides $\log(1+\mathrm{SNR}_2)$ bits of mutual information to decode the two messages. Thus decoding will be successful provided the rate is below (7.12).

Evidently, the basic structure of the coding scheme required is the same for the burst erasure channel with $\mathtt{T} = \mathtt{B}$ as for the AWGN fading channel with $\mathtt{T} = \mathtt{B}$ (although the rate obtained and decoding process are different). While the encoding scheme becomes more complicated for the general case of $\mathtt{M} > 1$ or $\mathtt{T} > \mathtt{B}$, this structural correspondence remains.

Sometimes it may be desirable to consider the minimum possible decoding delay as a function of rate as opposed to the maximum possible rate as a function of delay. This is possible by inverting (7.10) and so we obtain the following corollary.

**Corollary 6.** *The minimum possible ratio of decoding delay to burst length for a rate $R$, $\mathtt{M}$-path encoder which can correct any burst of length $\mathtt{B}$ with arbitrarily low maximal, block-based probability of error defined in* (7.6) *satisfies*

$$\frac{\mathtt{T}}{\mathtt{B}} = \min_{p_{\mathsf{x}}(x)} \max\left[\frac{1}{\mathtt{M}}\cdot\frac{\mathtt{M}\cdot I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})}{\mathtt{M}\cdot I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - R} - 1, 1\right] \tag{7.14a}$$

*for*

$$R \geq \mathtt{M}\cdot I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - \left[I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})\right] \tag{7.14b}$$

*and is zero otherwise.*

## ■ 7.4 The Converse of the Burst-Delay Capacity Theorem

In this section we prove the converse part of the coding theorem in Theorem 21.

**Theorem 22.** *Consider a pair of input compatible channel laws. An $\mathtt{M}$-path encoder which can correct any burst of length $\mathtt{B}$ after decoding delay $\mathtt{T}$ with arbitrarily low maximal, block-based probability of error has a rate less than the burst-delay capacity.*

*Proof.* Assume that the first $\mathtt{B}$ transmitted packets in path 0 suffer a burst starting at time 0, and the next $\mathtt{T}$ packets are successfully received in the good channel state. At this point the decoder must be able to decode everything up to and including $\mathbf{s}[\mathtt{B}]$. Therefore, if the next $\mathtt{B}$ packets in path 0 suffer a burst, they can be recovered once another $\mathtt{T}$ packets are successfully received in the good channel state. Repeating this pattern, we see that the decoder can completely recover from the alternating bursts in Fig. 7-2 each time $\mathtt{T}$ packets are received in the good channel state following a burst of $\mathtt{B}$ packets in the bad channel state.

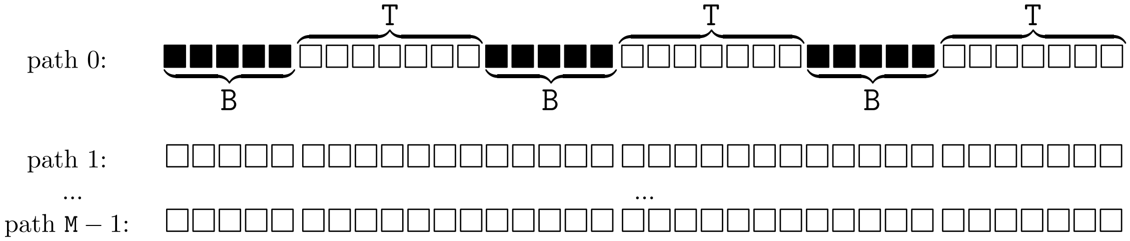


**Figure 7-2.** An encoder enabling correction of all bursts of length $\mathtt{B}$ on an arbitrary path with decoding delay $\mathtt{T}$ must allow correction of the displayed alternating burst pattern where shaded boxes indicate packets sent over a channel in the bad state.

If $L$ blocks of the alternating burst pattern occur then exactly $L \cdot [\mathtt{T} \cdot \mathtt{M} + (\mathtt{M}-1)\mathtt{B}]$ of the $L \cdot (\mathtt{T}+\mathtt{B}) \cdot \mathtt{M}$ packets sent arrive over the good channel while the remaining $L \cdot \mathtt{B}$ arrive over the bad channel. By Fano's inequality, the packets arriving over the good channel can each transmit at most $I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{G}})$ bits per packet while the packets arriving over the bad channel can transmit at most $I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{B}})$ bits per packet. Consequently, the total bit rate received must be less than

$$\frac{L \cdot [\mathtt{T} \cdot \mathtt{M} + (\mathtt{M}-1)\mathtt{B}]}{L \cdot (\mathtt{T}+\mathtt{B})} \cdot I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{G}}) + \frac{L \cdot \mathtt{B}}{L \cdot (\mathtt{T}+\mathtt{B})} \cdot I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{B}})$$

which reduces to (7.10) for the $\mathtt{T} \geq \mathtt{B}$ case.

The $\mathtt{T} < \mathtt{B}$ case, however, requires special treatment. Once again we assume that the first $\mathtt{B}$ packets transmitted in path 0 suffer a burst starting at time 0. If $\mathbf{s}[0]$ (the initial source packet) can be decoded at time $\mathtt{T}$, then at time $\mathtt{T}$ the recovered symbol stream is equivalent to what would be observed if a burst of $\mathtt{B}-1$ started at time 1. In other words, since the first packet lost in the burst can be recovered at time $\mathtt{T}$, after this recovery takes place it is as if $\mathbf{x}[0,0]$ was transmitted over the good channel state. Hence, if the next packet on path 0, $\mathbf{x}[\mathtt{B},0]$, arrives over the bad channel, then decoding can proceed as if a burst of length $\mathtt{B}$ had occurred starting at time 1. At time $\mathtt{T}+1$, $\mathbf{s}[1]$ will be recovered and the symbol stream will be equivalent to what would be observed if a burst of $\mathtt{B}-1$ started at time 2. Hence, if the next packet on path 0, $\mathbf{x}[0,\mathtt{B}+1]$, arrives over the bad channel, decoding can still proceed as if a burst of length $\mathtt{B}$ had occurred at time 2. By induction, this burst extending argument can be used to show that if a code can decode all bursts of length $\mathtt{B}$ with delay $\mathtt{T} < \mathtt{B}$, then it can decode when any path arrives over the bad channel for all time. By Fano's inequality, this implies the $\mathtt{M}-1$ good paths contribute $I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{G}})$ bits each, while the bad path contributes $I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{B}})$ bits. This establishes the second case in (7.10). □

## ■ 7.5 The Direct Part of the Burst-Delay Capacity Theorem

Since Theorem 22 provides an upper bound on the achievable rate, a natural question is whether this bound is achievable. In [119], a class of practical, single path, burst erasure correcting codes are presented which meet the lower bound with equality for certain rates.

To study the multiple path case for arbitrary channels, we consider a random coding construction and demonstrate that the bound in (7.10) is achievable. Our argument works for arbitrary channel laws since the the input compatible assumption is only required to show that (7.10) cannot be exceeded. While reading this section, the reader may find it useful to refer to the examples in Section 7.6 which provide more concrete single-path descriptions of codes for delay $\mathtt{T} = 3$ and burst length $\mathtt{B} = 2$.

## 7.5.1 Coding Scheme Overview

We begin by providing an intuitive explanation of our coding scheme. Essentially, coding works by using a combination of interleaving and binning to spread the information in each source packet into later packets. For example, one possible interleaving scheme would be to divide each source packet, $\mathbf{s}[i]$, into thirds, repeat the first third in packets $i+1$ and $i+2$, and repeat the remaining two-thirds in packet $i+3$. If we denote source packets as consisting of 3 components

$$\mathbf{s}[i] = (\mathbf{s}_0[i], \mathbf{s}_1[i], \mathbf{s}_2[i]),$$

then the example interleaving schedule can be represented as

$$\mathbf{x}[i] = (\mathbf{s}_0[i], \mathbf{s}_1[i], \mathbf{s}_2[i], \mathbf{s}_0[i-1], \mathbf{s}_0[i-2], \mathbf{s}_1[i-3], \mathbf{s}_2[i-3]). \tag{7.15}$$

If an erasure burst of length $\mathtt{B} = 2$ starts at time $j$, it is easy to verify that the lost data can be recovered with a decoding delay of $\mathtt{T} = 3$ from the repeated data.

Unfortunately, this system has a rate of $R = 3/7$ while, according to (7.10), a rate 3/5 system should suffice. Evidently, purely repeating data and interleaving is inefficient: it requires too much redundancy. To reduce the amount of redundancy required we combine interleaving with binning. Specifically, we first repeat and interleave to obtain an intermediate representation, $\mathbf{v}[i]$, for each channel input to send at time $i$ on path $p$. For example, the intermediate representation

$$\mathbf{v}[i] = (\mathbf{s}_0[i], \mathbf{s}_1[i], \mathbf{s}_2[i], \mathbf{s}_0[i-1], \mathbf{s}_0[i-2], \mathbf{s}_1[i-3], \mathbf{s}_2[i-3]) \tag{7.16}$$

corresponds to (7.16). Instead of sending the full intermediate representation, however, we partition all possible values for $\mathbf{v}[i]$ into bins and send only the bin number. Thus the transmitted packet is denoted

$$\mathbf{x}[i] = \mathcal{H}(\mathbf{v}[i]) \tag{7.17}$$

where $\mathcal{H}(\cdot)$ denotes the bin number assigned to the input. By choosing appropriate interleaving and binning schemes, we will show it is possible to obtain an efficient system (*i.e.*, one which satisfies (7.10)).

## 7.5.2 Encoding

For $\mathtt{T} < \mathtt{B}$, using a standard random code construction to encode the source data, $\mathbf{s}[i]$, into the $\mathtt{M}$ packets, $\mathbf{x}[i,p]$, transmitted through the channel is sufficient. This is because each of the $\mathtt{M}-1$ packets received over the good channel law contributes about $n_c \cdot I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta} = \theta_{\mathcal{G}})$ bits while the packet received over the bad channel law conveys about $n_c \cdot I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta} = \theta_{\mathcal{B}})$ bits.

Hence, we focus mostly on the case with $\mathtt{T} \geq \mathtt{B}$. Specifically, we describe time and path invariant interleaving and binning schedules for a rate $R$ system operating over $\mathtt{M}$ paths designed to correct bursts of length $\mathtt{B}$ with decoding delay $\mathtt{T} \geq \mathtt{B}$.

### Dividing Source Packets

First, we divide each source packet at time $i$, $\mathbf{s}[i]$, into $\mathtt{M}$ pieces and associate each piece with one possible path to obtain $\mathbf{s}[i,p]$ with $p \in \{0,1,\ldots,\mathtt{M}-1\}$. Next we further sub-divide each $\mathbf{s}[i,p]$ into $\mathtt{L}$ pieces denoted $\mathbf{s}_j[i,p]$ with

$$\mathtt{L} \triangleq \mathtt{T}\cdot\mathtt{M}+\mathtt{B}\cdot(\mathtt{M}-1)+1. \tag{7.18}$$

We place

$$n_c \cdot [I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})-\epsilon] \tag{7.19}$$

bits in the last piece, $\mathbf{s}_{\mathtt{L}-1}[i,p]$, for each path and evenly distribute the rest of the source data among the remaining pieces yielding

$$n_c \cdot \frac{C_{\mathtt{B},\mathtt{T}} - \mathtt{M}\cdot I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})-\epsilon}{\mathtt{L}-1} \tag{7.20}$$

bits for each piece $\mathbf{s}_j[i,p]$ with $j \in \{0,1,\ldots,\mathtt{L}-2\}$. This splitting scheme is summarized in Table 7.1.

**Table 7.1.** Each source packet is split into $\mathtt{M}\cdot\mathtt{L}$ pieces. The number of bits for the last piece on each path is specified by (7.19) and the size for the remaining pieces are specified in (7.20).

$$\mathbf{s}[i] \triangleq \begin{pmatrix} \mathbf{s}_0[i,0] & \mathbf{s}_1[i,0] & \ldots & \mathbf{s}_{\mathtt{L}-1}[i,0] \\ \mathbf{s}_0[i,1] & \mathbf{s}_1[i,1] & \ldots & \mathbf{s}_{\mathtt{L}-1}[i,1] \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{s}_0[i,\mathtt{M}-1] & \mathbf{s}_1[i,\mathtt{M}-1] & \ldots & \mathbf{s}_{\mathtt{L}-1}[i,\mathtt{M}-1] \end{pmatrix}$$

### Interleaving the Divided Packets

The pieces of each source packet are then interleaved to produce the intermediate representation $\mathbf{v}[i,p]$ as specified in Table 7.2 and discussed in more detail below. The interleaving schedule is easiest to understand if we imagine that the transmitter "knows" a burst of length $\mathtt{B}$ will start at time $i$ on path $p$ and designs an interleaver to allow quick recovery. Specifically, only $\mathbf{s}[i,p]$, $\mathbf{s}[i+1,p]$, …, $\mathbf{s}[i+\mathtt{B}-1,p]$ are affected by the burst, therefore these are the components considered by the interleaver inside the outermost for loop in Table 7.2.

Packets which follow the burst can communicate at most $I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}})$ bits of information while packets damaged by the burst can communicate at most $I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})$ bits. One approach to coding is to choose an interleaving schedule such that the receiver can recover $\mathbf{s}_{\mathtt{L}-p-1}[i]$ from $y[i,p]$, and recover the remaining $\mathtt{L}-1$ pieces from the packets not suffering the burst. But how many pieces of $\mathbf{s}[i]$ should be placed in later packets?

To answer this question, note that for a system transmitting at the burst-delay capacity, each of the packets transmitted on the $\mathtt{M}$ possible paths contains $n_c C_{\mathtt{B},\mathtt{T}}/\mathtt{M}$ bits of new information. Thus if a packet is received over the good channel state after the burst, the difference between the mutual information and the amount of new information sent is

**Table 7.2.** Description of how $\mathbf{s}\,[i]$'s are interleaved to produce $\mathbf{v}[i,p]$'s.

```
for $p \in \{0, 1, \ldots, \mathtt{M}-1\}$ do
  for $l \in \{0, 1, \ldots, \mathtt{L}-1\}$ do
    Put $\mathbf{s}_l[i,p]$ in $\mathbf{v}[i,p]$
  end for
  for $b \in \{0, 1, \ldots, \mathtt{B}-1\}$ do
    for $m \in \{0, 1, \ldots, \mathtt{M}-1\}$ do
      $j \leftarrow b + m\mathtt{B}$
      $i' \leftarrow i + \mathtt{T}$
      $p' \leftarrow p + m \mod \mathtt{M}$
      Put $\mathbf{s}_j[i,p]$ in $\mathbf{v}[i',p']$
    end for
  end for
  for $t \in \{0, 1, \ldots, \mathtt{T}-\mathtt{B}-1\}$ do
    $j \leftarrow \mathtt{BM} + t$
    $i' \leftarrow i + t$
    for $b \in \{0, 1, \mathtt{B}-1\}$ do
      Put $\mathbf{s}_j[i,p]$ in $\mathbf{v}[i'+1+b,p]$
    end for
  end for
  for $t \in \{0, 1, \ldots, \mathtt{T}-1\}$ do
    for $m \in \{0, 1, \ldots, \mathtt{M}-2\}$ do
      $j \leftarrow \mathtt{B}(\mathtt{M}-1) + \mathtt{T} + t + m\mathtt{T}$
      $i' \leftarrow i + t$
      $p' \leftarrow p + 1 + m \mod \mathtt{M}$
      for $b \in \{0, 1, \mathtt{B}-1\}$ do
        Put $\mathbf{s}_j[i,p]$ in $\mathbf{v}[i'+b,p']$
      end for
    end for
  end for
end for
```

roughly

$$n_c \cdot [I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{G}}) - C_{\mathtt{B},\mathtt{T}}/\mathtt{M}] = n_c \cdot \frac{\mathtt{B}}{(\mathtt{T}+\mathtt{B})\cdot\mathtt{M}} \cdot [I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{G}}) - I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{B}})]. \quad (7.21)$$

Thus, the amount of information about packets affected by the burst which can be recovered by a packet following the burst is exactly (7.21). Stated differently, each packet can carry about $n_c \cdot C_{\mathtt{B},\mathtt{T}}/\mathtt{M}$ bits of new information in addition to the amount specified by (7.21) which is used to carry redundant information about past data. For $\mathtt{T} > \mathtt{B}$, each packet sent at time $i + \mathtt{T}$ will be received regardless of which path it takes. Therefore, we can start by placing the maximum amount of redundant information from $\mathbf{s}[i]$ in the transmitted packets at time $i + \mathtt{T}$. Dividing the room left for redundant information in each packet (7.21) by the size of pieces 0 through $\mathtt{L} - 2$ (7.20) indicates that $\mathtt{B}$ pieces can be placed in each packet at time $i + \mathtt{T}$. This explains step 10 of Table 7.2.

Next, since the transmitter "knows" packets $\mathbf{x}[i,p]$ through $\mathbf{x}[\mathtt{B}-1,p]$ will be lost in the burst, there are $\mathtt{T} + 1 - \mathtt{B}$ packets remaining on path $p$ to which we can interleave information. Therefore we interleave one piece of redundant information to each such $\mathbf{v}[i,p]$ input in step 17. (The loop in step 16 also interleaves the same piece to each of the preceding $\mathtt{B}$ packets $\mathbf{v}[i,p]$, but this will be discussed later). Similarly, there are $(\mathtt{M}-1)(\mathtt{T}+\mathtt{B})$ packets on the paths not affected by the burst within the decoding delay window which have not yet been assigned redundant information. Thus we assign each of these a different redundant piece in step 26. (Again the loop in step 25 also interleaves the same piece to each of the preceding $\mathbf{v}[i,p]$). This accounts for all pieces of $\mathbf{s}[i,p]$ except $\mathbf{s}_{\mathtt{L}-1}[i,p]$ which is not interleaved at all.

Of course, the transmitter knows neither when the burst will start nor what path it will strike. Therefore, the encoder essentially repeats the previous interleaving scheme with $\mathtt{B}$ possible time shifts and $\mathtt{M}$ possible path shifts. The time shifts explain the for loops in steps 16 and 25 mentioned earlier while the path shift explains the outermost for loop.

### Binning the Interleaved Pieces

The interleaving schedule described above requires too much redundancy to be efficient by itself. Therefore, instead of sending each $\mathbf{v}[i,p]$, we send only the bin index of each packet. Specifically, we define a hash partition with output length $N$ as a mapping

$$\mathcal{H}(\cdot) : \mathcal{S}^* \mapsto \{0,1\}^N \quad (7.22)$$

which takes as input an arbitrary length sequence of source symbols from the alphabet $\mathcal{S}$ and returns $N$ bits. Instead of describing the a hash partition with a specific input–output relationship, we work with random hash partitions where every possible input is assigned a random $N$-bit output.[3] To construct burst-delay capacity achieving systems, we use random hash partitions with output length

$$n_c \cdot [I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{G}}) - \epsilon]/\mathtt{M} \quad (7.23)$$

and transmit the channel input $\mathbf{x}[i,p]$ by selecting codeword number $\mathcal{H}(\mathbf{v}[i,p])$ from a random channel codebook.

[3] By this we mean that the input–output relationship is randomly constructed *not* that $\mathcal{H}(s)$ may produce different random values when called with the same fixed input $s$.

### Encoder Summary

To summarize, the encoding process is as follows. For each time $i$ and each path $p$, a rate $2^{n_c(R)/\mathtt{M}}$, length $n_c$, random codebook is constructed by selecting $2^{n_c(R)/\mathtt{M}}$ sequences of length $n_c$ according to the i.i.d. distribution $p_{\mathsf{s}}(\cdot)$. Each source packet, $\mathbf{s}\,[i]$ is divided and interleaved as described in Tables 7.1 and 7.2 to produce the intermediate representations $\mathbf{v}[i,p]$. A bin index, $\mathcal{H}(\mathbf{v}[i,p])$ is computed and the corresponding codeword in the codebook for time $i$ and path $p$ is transmitted through the channel.

## ■ 7.5.3 Decoding

To decode $\mathbf{s}\,[i]$, the receiver first decodes the bin indexes from the channel output and then decodes the pieces of the source from the bin indexes. For packets received over the good channel state, decoding the bin index for $\mathbf{x}[i,p]$ is accomplished by looking for a codeword jointly typical with the channel output $\mathbf{y}[i,p]$. This will succeed with high probability according to standard arguments since the transmitted rate is less than the mutual information. Furthermore, since the codebooks at each time and path are independent there are no error propagation or other dynamic issues in decoding the bin indexes.

Therefore, the main effort in describing and analyzing the decoding process is in decoding the pieces of the source packets from the bin indexes. Essentially, the decoder accomplishes this by trying to find a sequence of source packets jointly typical with the decoded bin indexes. If a unique source packet is jointly typical it is declared to be $\mathbf{s}\,[i]$, otherwise a decoding error occurs. For simplicity, we consider a suboptimal bin decoding process which nonetheless achieves capacity.

The analysis of the decoding process is somewhat lengthy and complicated. Hence the reader may find it useful to read the examples in Section 7.6 either before or in concert with the rest of this section.

### Decoding When No Burst Occurs

**Lemma 1.** *When no burst occurs, $\mathbf{s}[i,p]$ can be decoded from $\mathbf{y}[i,p]$ with probability of error tending to zero as $n_c \to 0$.*

*Proof.* Assume for the time being that all past source packets have been successfully recovered. With this assumption, decoding $\mathbf{s}\,[i]$ from the bin indexes obtained from $\mathbf{x}[i,p]$ for $p \in \{0,1,\ldots,\mathtt{M}-1\}$ is straightforward. Specifically, while each $\mathbf{v}[i,p]$ includes data from current and past source packets, the past source data in $\mathbf{v}[i,p]$ is known by assumption and can be ignored. Thus the unknown pieces are exactly the $R \cdot n_c$ bits of $\mathbf{s}\,[i]$. Since we have $\mathtt{M}$ bin indexes each consisting of $n_c \cdot I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}})-\epsilon]/\mathtt{M}$ bits, the probability that an incorrect source packet will be jointly typical with all the bin indexes is exactly

$$2^{n_c \cdot R} \cdot 2^{-n_c \cdot I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}})-\epsilon]} = 2^{-n_c[I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}})-\epsilon-R]} \tag{7.24}$$

which goes to zero asymptotically with $n_c$ for any $\epsilon > 0$ provided $R \leq C_{\mathtt{B},\mathtt{T}}$. The correct source packet will always be jointly, and hence the probability of decoding error given the past source data was correctly decoded and the current channel output was received over the good channel state is approximately (7.24).[4]

[4] We say approximately and not exactly since the true probability of error must also take into account the possibility that the bin indexes are incorrectly decoded from the channel outputs. For $R \leq C_{\mathtt{B},\mathtt{T}}$, this probability goes to zero so our approximation is justified. A sharper analysis, *e.g.*, one which analyzes error exponents, would need to take this into account, however.

Finally, we address the assumption that all past source packets have been successfully decoded. This is trivially true for the initial packet and therefore our argument shows the first packet can be successfully decoded with some small error probability $\epsilon'$. Thus our assumption is true for the second packet which can be decoded with a probability of error at most $2\epsilon'$. Continuing by induction shows that the error probability for the $n$th packet is no more than $n \cdot \epsilon'$. Thus the error probability for all the packets can be made negligible provided that $\epsilon'$ decreases faster than $n$.

One way to accomplish this is to make sure that the packet size $n_c$ grows fast enough (and hence the error probability $\epsilon'$ drops fast enough) with $n$ to ensure that the product $n\epsilon'$ goes to zero. Increasing $n_c$ with $n$ may be sufficient to provide simple bounds on the error probability when $n$ is finite, but it is somewhat undesirable if $n$ is large or even infinite. In general, it is possible to obtain an error probability $\epsilon'$ which depends only on $n_c$ and is independent of $n$. This is achieved either by a detailed analysis of error propagation or by considering a more complicated decoding scheme. For the latter, the receiver decodes $\mathbf{s}[i]$ by looking not only at the current channel output $\mathbf{y}[i]$ but at all past channel outputs. For such decoding, the difference between the received mutual information and the transmitted data increases with both $n_c$ and $n$ and so the error probability does not grow with $n$. We omit such an analysis since our focus is on the first order effect of delay. Understanding the details of the probability of error is an interesting topic for future work, however, and might have connections to [157].

□

#### Decoding When a Burst Occurs

Next we assume that decoding was successful before time $i$ when a burst of length $\mathtt{B}$ started on path $p$. At first we give up on decoding the packets, $\mathbf{y}[i,p]$, $\mathbf{y}[i+1,p]$, …, $\mathbf{y}[i+\mathtt{B}-1,p]$, which are affected by the burst and declare the corresponding bin indexes $\mathcal{H}(\mathbf{v}[i,p])$, $\mathcal{H}(\mathbf{v}[i+1,p])$, …, $\mathcal{H}(\mathbf{v}[i+\mathtt{B}-1,p])$ to be erased. The other bin indexes can still be successfully decoded, however, according to standard arguments.

Our main tool in decoding the intermediate representations $\mathbf{v}[i,p]$ from the successfully recovered bin indexes is the following lemma proved in Appendix E.

**Lemma 2.** *Consider a bin index, $\mathcal{H}(\mathbf{v}[i,p])$, where the only unknown source components are $\mathbf{s}[i,p]$ and at most $\mathtt{B}$ redundant pieces of the form $\mathbf{s}_b[a,c]$ with $0 \leq b \leq \mathtt{L}-2$ of size* (7.20)*. All of these source components can be recovered with error probability approaching zero as $n_c \to \infty$.*

The intuition for this result is that each packet received over the good channel state can convey about $n_c \cdot I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{G}})$ bits. Piece $\mathbf{s}[i,p]$ takes about $n_c \cdot C_{\mathtt{B},\mathtt{T}}$ bits while each of the other unknown components has a size of about (7.20). Thus, as discussed in Section 7.5.2, $n_c \cdot I(x;y|\boldsymbol{\theta} = \theta_{\mathcal{G}})$ will be larger than the amount of data in the unknown source components so an appropriately encoded system will be able to decode correctly.

We begin by focusing on the $\mathtt{B} \cdot (\mathtt{M}-1)$ bin indexes which are "parallel" with the burst but not affected by it.

**Lemma 3.** *Imagine that all source data packets are correctly decoded until a burst starts at time $i$ on path $p$ and let $p' \neq p$ and $t \in \{0, 1, \ldots, \mathtt{B}-1\}$. Then all received packets of the form $\mathbf{y}[i+t,p']$ can be decoded to their bin indexes $\mathcal{H}(\mathbf{v}[i+t,p'])$ and the corresponding bin indexes can be decoded to their intermediate representations $\mathbf{v}[i+t,p']$ with negligible probability of error. Furthermore, the decoding delay is at most $\mathtt{T}$.*

*Proof.* The rate of each bin index is chosen to be less than the channel mutual information $I(x; y|\boldsymbol{\theta} = \theta_{\mathcal{G}})$. Therefore, the bin indexes $\mathcal{H}(\mathbf{v}[i+t, p'])$ can be recovered using standard arguments. The unknown components in each bin index are $\mathbf{s}[i+t, p']$ as well as at most $\mathtt{B}$ other pieces of the form $\mathbf{s}_b[a, c]$ with $0 \leq b \leq \mathtt{L} - 2$. This follows by construction and specifically by Table 7.2.

In particular, only one component of $\mathbf{s}[i, p]$ is placed in each $\mathbf{v}[i, p']$ in step 26 of Table 7.2 so Lemma 2 can be applied to show that is $\mathcal{H}(\mathbf{v}[i, p'])$ decoded successfully. Similarly, only one component of $\mathbf{s}[i+1, p]$ is placed in each $\mathbf{v}[i+1, p']$. Note, however, that a new component of $\mathbf{s}[i, p]$ is also placed in $\mathbf{v}[i+1, p']$ along with the component of $\mathbf{s}[i, p]$ which was placed in $\mathbf{v}[i, p']$ due to the loop in step 25 of Table 7.2. Since this last component was recovered when $\mathcal{H}(\mathbf{v}[i, p'])$ was decoded in the previous step we can treat it as known when decoding $\mathcal{H}(\mathbf{v}[i+1, p'])$. Thus we can again apply Lemma 2 to show $\mathcal{H}(\mathbf{v}[i+1, p'])$ is decoded.

In general, $\mathbf{v}[i+t, p']$ contains $t+1$ components from $\mathbf{s}[i, p]$, plus $t$ components from $\mathbf{s}[i+1, p]$, plus $t-1$ components from $\mathbf{s}[i+2, p]$, and so on down to one component from $\mathbf{s}[i+t, p]$ for a total of $(t+2)(t+1)/2$ components. The key insight of this lemma is that many of these components are repeated. Specifically, of the two components from $\mathbf{s}[i+t-1, p]$ in $\mathbf{v}[i+t, p']$, one of these was recovered when $\mathcal{H}(\mathbf{v}[i+t-1, p'])$ was decoded. Similarly, of the three components from $\mathbf{s}[i+t-2, p']$ in $\mathbf{v}[i+t, p']$, two were recovered when $\mathcal{H}(\mathbf{v}[i+t-1, p'])$ was decoded, and so on. Thus, $\mathbf{v}[i+t, p']$ contains at most $t$ unknown source components. Since $t \leq \mathtt{B} - 1$ by assumption, Lemma 2 implies that the desired intermediate representations can be decoded. □

Next we consider the $(\mathtt{T} + 1 - \mathtt{B}) \cdot \mathtt{M}$ packets following the burst.

**Lemma 4.** *Imagine that all source data packets are correctly decoded until a burst starts at time $i$ on path $p$ and let $t \in \{0, 1, \ldots, \mathtt{T}+1-\mathtt{B}\}$ and $p' \in \{0, 1, \ldots, \mathtt{M}-1\}$. Then all received packets of the form $\mathbf{y}[i+\mathtt{B}+t, p']$ can be decoded to their bin indexes $\mathcal{H}(\mathbf{v}[i+\mathtt{B}+t, p'])$ and the corresponding bin indexes can be decoded to their intermediate representations $\mathbf{v}[i+t, p']$ with negligible probability of error. Furthermore, the decoding delay is at most $\mathtt{T}$.*

*Proof.* As in Lemma 3, each bin index can be decoded and the unknown components in each $\mathbf{v}[i+t, p']$ contain components of the form $\mathbf{s}[i+t, p']$ as well as at most $\mathtt{B}$ other pieces of the form $\mathbf{s}_b[a, c]$ with $0 \leq b \leq \mathtt{L}-2$. Establishing this is slightly different than in, Lemma 3, however.

For all but the last packet, (*i.e.*, for all $t \in \{0, 1, \ldots, \mathtt{T}+1-\mathtt{B}\}$ except $t = \mathtt{T}+1-\mathtt{B}$) the situation is essentially the same as for decoding the last set of packets in Lemma 3. Specifically, each bin index $\mathcal{H}(\mathbf{v}[i+t, p'])$ contains exactly $\mathtt{B}$ unknown source components in addition to $\mathbf{s}[i+t, p']$ because other source components from $\mathbf{s}[i, p]$, $\mathbf{s}[i, p]$, …, $\mathbf{s}[i+\mathtt{B}-1, p]$ in $\mathcal{H}(\mathbf{v}[i+t, p'])$ have been recovered in previous steps.[5]

The last set of bin indexes, $\mathcal{H}(\mathbf{v}[i+\mathtt{T}-1, p'])$, each contain exactly $\mathtt{B}$ components from $\mathbf{s}[i, p]$ added in step 10 of Table 7.2 along with $\mathbf{s}[i+\mathtt{T}-1, p']$ and previously recovered source components. Thus Lemma 2 implies that successful decoding occurs except with negligible probability. □

[5] The situation is essentially the same and not exactly because in Lemma 3 going from time $i+a$ to $i+a+1$ always added more unknown source components to the bin index to be decoded. In Lemma 4, however, going from time $i+t$ to $i+t+1$ results in the same number of unknown source components in the bin index to be decoded. Also, only paths with $p' \neq p$ were considered in the former while all paths are considered in the latter.

The previous lemmas establish successful decoding for all components of $\mathbf{s}_j[i,p]$ with $j \leq \mathtt{L}-1$, while the following considers the last piece.

**Lemma 5.** *Imagine that all source data packets are correctly decoded until a burst starts at time $i$ on path $p$. Then $\mathbf{s}_{\mathtt{L}-1}[i,p]$ can be decoded with negligible probability of error with a decoding delay of* $\mathtt{T}$.

*Proof.* According to the previous lemmas, after a delay of $\mathtt{T}$ (*i.e.*, at time $i+\mathtt{T}+1$), all the components of $\mathbf{s}[i,p]$ can be successfully decoded except possibly $\mathbf{s}_{\mathtt{L}-1}[i,p]$ which we now consider. By construction in (7.19), $\mathbf{s}_{\mathtt{L}-1}[i,p]$ is of size $n_c \cdot [I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})-\epsilon]$ bits. Therefore, there are

$$2^{n_c \cdot [I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})-\epsilon]} \tag{7.25}$$

possible values for $\mathbf{s}_{\mathtt{L}-1}[i,p]$ and similarly, the number of possibilities for $\mathbf{x}[i,p]$ is also (7.25). The decoder searches for an $\mathbf{x}[i,p]$ which is jointly typical with the received channel output $\mathbf{y}[i,p]$. The correct codeword will be typical with high probability, while the probability that a given incorrect codeword is typical is roughly $2^{-n_c I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})}$. Thus the probability that any incorrect codeword is typical is about

$$2^{n_c \cdot [I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})-\epsilon]} \cdot 2^{-n_c I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})} = 2^{-n_c \epsilon}. \tag{7.26}$$

Thus decoding is successful with asymptotically negligible probability of error. □

Combining all our results for this section yields the following theorem.

**Theorem 23.** *Any rate less than the burst-delay capacity in* (7.10) *is achievable with decoding delay* $\mathtt{T}$.

*Proof.* Lemmas 3, 4, and 5 imply that $\mathbf{s}[i,p]$ can be recovered with delay $\mathtt{T}$ when a burst of length $\mathtt{B}$ starts at time $i$ on an arbitrary path $p$. Since the encoding scheme used is time invariant, the same argument shows that all packets affected by the burst can also be recovered with delay $\mathtt{T}$. □

## ■ 7.6 Single Path Examples

### ■ 7.6.1 An Erasure Channel Example

Imagine that we which to design a system to correct for erasure bursts of length $\mathtt{B}=2$, with decoding delay $\mathtt{T}=3$, using $\mathtt{M}=1$ path. Specifically, the good channel corresponds to a noiseless binary channel (*i.e.*, $I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{G}})=1$) and the bad channel is a binary erasure channel, (*i.e.*, $I(\mathsf{x};\mathsf{y}|\boldsymbol{\theta}=\theta_{\mathcal{B}})=0$). According to (7.10), the burst-delay capacity is 3/5. Thus if each packet transmitted through the channel contains $n_c$ bits, then about $(3/5)n_c$ source bits per packet can be reliably communicated.

Based on the coding scheme described in Section 7.5.2 each source packet should be divided into $\mathtt{L}=4$ pieces, but since according to (7.19) the last piece is empty we ignore it in this example. Thus each $\mathbf{s}[i]$ is divided into the three pieces $(\mathbf{s}_0[i], \mathbf{s}_1[i], \mathbf{s}_2[i])$ with each piece consisting of about $n_c/5$ bits. Applying the prescription in Section 7.5.2 and in particular the interleaving schedule in Table 7.2 produces the packet stream shown in Table 7.3.

If all packets before time $i$ are received correctly and $\mathbf{x}[i]$ and $\mathbf{x}[i+1]$ are lost in burst of length 2, decoding proceeds as follows. At time $i+2$, the decoder receives $\mathbf{x}[i+2]$ and

**Table 7.3.** Encoding example for a rate $R = 3/5$ single path code. Each source component is split into 3 equal pieces and interleaved. Instead of transmitting the interleaved data directly, a binning function $\mathcal{H}_{5/7}(\cdot)$ maps $(7/5-\epsilon)\cdot n_c$ input bits to a bin number consisting of $n_c$ bits.

$$
\begin{aligned}
\mathbf{x}[i-1] &= \mathcal{H}_{5/7}\big(\mathbf{s}_0[i-1], \mathbf{s}_1[i-1], \mathbf{s}_2[i-1], \mathbf{s}_0[i-2], \mathbf{s}_0[i-3], \mathbf{s}_1[i-4], \mathbf{s}_2[i-4]\big)\\
\mathbf{x}[i+0] &= \mathcal{H}_{5/7}\big(\mathbf{s}_0[i+0], \mathbf{s}_1[i+0], \mathbf{s}_2[i+0], \mathbf{s}_0[i-1], \mathbf{s}_0[i-2], \mathbf{s}_1[i-3], \mathbf{s}_2[i-3]\big)\\
\mathbf{x}[i+1] &= \mathcal{H}_{5/7}\big(\mathbf{s}_0[i+1], \mathbf{s}_1[i+1], \mathbf{s}_2[i+1], \mathbf{s}_0[i+0], \mathbf{s}_0[i-1], \mathbf{s}_1[i-2], \mathbf{s}_2[i-2]\big)\\
\mathbf{x}[i+2] &= \mathcal{H}_{5/7}\big(\mathbf{s}_0[i+2], \mathbf{s}_1[i+2], \mathbf{s}_2[i+2], \mathbf{s}_0[i+1], \mathbf{s}_0[i+0], \mathbf{s}_1[i-1], \mathbf{s}_2[i-1]\big)\\
\mathbf{x}[i+3] &= \mathcal{H}_{5/7}\big(\mathbf{s}_0[i+3], \mathbf{s}_1[i+3], \mathbf{s}_2[i+3], \mathbf{s}_0[i+2], \mathbf{s}_0[i+1], \mathbf{s}_1[i+0], \mathbf{s}_2[i+0]\big)\\
\mathbf{x}[i+4] &= \mathcal{H}_{5/7}\big(\mathbf{s}_0[i+4], \mathbf{s}_1[i+4], \mathbf{s}_2[i+4], \mathbf{s}_0[i+3], \mathbf{s}_0[i+2], \mathbf{s}_1[i+1], \mathbf{s}_2[i+1]\big)
\end{aligned}
$$

searches for a collection of pieces

$$(\mathbf{s}_0[i+2], \mathbf{s}_1[i+2], \mathbf{s}_2[i+2], \mathbf{s}_0[i], \mathbf{s}_0[i+1]) \tag{7.27}$$

which has a bin number corresponding to $\mathbf{x}[i+2]$. The correct collection will always have the correct bin number. The probability that an incorrect collection has the correct bin number is $2^{-n_c}$. Since the correct value for $\mathbf{s}[i-1]$ is known and each unknown source piece consists of about $n_c R/3$ bits, there are $2^{(5/3)n_c R}$ possible collections. Thus the expected number of incorrect collections which nonetheless have the correct bin number is

$$2^{-n_c} \cdot 2^{n_c(5/3)R} = 2^{-n_c[1-(5/3)R]}.$$

Thus provided that $R < 3/5 - \epsilon$, the expected number of incorrect collections which have the correct bin number is at most $2^{-n_c(5/3)\epsilon}$ and the probability of decoding failure goes to zero as $n_c \to \infty$.

Similarly, when $\mathbf{x}[i+3]$ is received the decoder searches for a collection of pieces

$$(\mathbf{s}_0[i+3], \mathbf{s}_1[i+3], \mathbf{s}_2[i+3], \mathbf{s}_0[i+1], \mathbf{s}_0[i+2], \mathbf{s}_1[i], \mathbf{s}_2[i])$$

which has a bin number matching $\mathbf{x}[i+3]$. Provided that $\mathbf{s}_0[i+1]$ and $\mathbf{s}[i+2]$ were correctly determined in the preceding step, the probability that an incorrect collection has the correct bin number will be small according to the previous argument. Thus, with probability arbitrarily close to 1, the decoder will correctly recover the lost packets.

## 7.6.2 A Single Path Binary Symmetric Channel Example

Next consider a noiseless binary channel as the good channel and a binary symmetric channel with cross-over probability $p$ as the bad channel. If $p$ is chosen such that $H_b(p) = 1/3$ where $H_b(\cdot)$ denotes the binary entropy function, then the only difference as compared to the erasure channel example is that now $I(x; y|\boldsymbol{\theta} = \theta_{\mathcal{B}}) = 1 - H_b(p) = 2/3$ and therefore each source packet is split into 4 pieces where the last piece contains $(2/3 - \epsilon) \cdot n_c$ bits and the remaining 3 pieces each contain $(1/15 - \epsilon) \cdot n_c$ bits. The total amount of bits in each source packet is

$$(3/15 + 2/3 - 4\epsilon)n_c = (13/15 - 4\epsilon)n_c \tag{7.28}$$

bits which equals the bound (7.10) for successful decoding of bursts of length $\mathtt{B} = 2$ with delay $\mathtt{T} = 3$:

$$C_{\mathtt{B},\mathtt{T}} = \frac{3}{5} + \frac{2}{5} \cdot (1 - H_b(p)) = \frac{3}{5} + \frac{2}{5} \cdot \frac{2}{3} = \frac{13}{15}.$$

Applying the interleaving schedule described in Table 7.2 produces the packet stream shown in Table 7.4.

**Table 7.4.** Encoding example for a burst binary symmetric channel. Each source packet is split into four pieces and interleaved. Instead of transmitting the interleaved data directly, a binning function, $\mathcal{H}_{15/17}(\cdot)$ maps $(17/15 - 8\epsilon)n_c$ input bits to a bin number consisting of $n_c$ bits.

$$\begin{aligned}
\mathbf{x}[i-1] &= \mathcal{H}_{15/17}\big(\mathbf{s}_0[i{-}1], \mathbf{s}_1[i{-}1], \mathbf{s}_2[i{-}1], \mathbf{s}_3[i{-}1], \mathbf{s}_0[i{-}2], \mathbf{s}_0[i{-}3], \mathbf{s}_1[i{-}4], \mathbf{s}_2[i{-}4]\big)\\
\mathbf{x}[i+0] &= \mathcal{H}_{15/17}\big(\mathbf{s}_0[i{+}0], \mathbf{s}_1[i{+}0], \mathbf{s}_2[i{+}0], \mathbf{s}_3[i{+}0], \mathbf{s}_0[i{-}1], \mathbf{s}_0[i{-}2], \mathbf{s}_1[i{-}3], \mathbf{s}_2[i{-}3]\big)\\
\mathbf{x}[i+1] &= \mathcal{H}_{15/17}\big(\mathbf{s}_0[i{+}1], \mathbf{s}_1[i{+}1], \mathbf{s}_2[i{+}1], \mathbf{s}_3[i{+}1], \mathbf{s}_0[i{+}0], \mathbf{s}_0[i{-}1], \mathbf{s}_1[i{-}2], \mathbf{s}_2[i{-}2]\big)\\
\mathbf{x}[i+2] &= \mathcal{H}_{15/17}\big(\mathbf{s}_0[i{+}2], \mathbf{s}_1[i{+}2], \mathbf{s}_2[i{+}2], \mathbf{s}_3[i{+}2], \mathbf{s}_0[i{+}1], \mathbf{s}_0[i{+}0], \mathbf{s}_1[i{-}1], \mathbf{s}_2[i{-}1]\big)\\
\mathbf{x}[i+3] &= \mathcal{H}_{15/17}\big(\mathbf{s}_0[i{+}3], \mathbf{s}_1[i{+}3], \mathbf{s}_2[i{+}3], \mathbf{s}_3[i{+}3], \mathbf{s}_0[i{+}2], \mathbf{s}_0[i{+}1], \mathbf{s}_1[i{+}0], \mathbf{s}_2[i{+}0]\big)\\
\mathbf{x}[i+4] &= \mathcal{H}_{15/17}\big(\mathbf{s}_0[i{+}4], \mathbf{s}_1[i{+}4], \mathbf{s}_2[i{+}4], \mathbf{s}_3[i{+}4], \mathbf{s}_0[i{+}3], \mathbf{s}_0[i{+}2], \mathbf{s}_1[i{+}1], \mathbf{s}_2[i{+}1]\big)
\end{aligned}$$

If all packets before time $i$ are received correctly and $\mathbf{x}[i]$ and $\mathbf{x}[i+1]$ are lost in burst of length 2, decoding proceeds as follows. At time $i+2$, the decoder receives $\mathbf{x}[i+2]$ and searches for a collection of pieces

$$(\mathbf{s}_0[i+2], \mathbf{s}_1[i+2], \mathbf{s}_2[i+2], \mathbf{s}_3[i+2], \mathbf{s}_0[i], \mathbf{s}_0[i+1]) \qquad (7.29)$$

which has a bin number corresponding to $\mathbf{x}[i+2]$. The correct collection will always have the correct bin number. The probability that a given incorrect collection has the same bin number as $\mathbf{x}[i+2]$ is $2^{-n_c}$. Since the correct value for $\mathbf{s}[i-1]$ is known, there are

$$2^{n_c \cdot (13/15 - 4\epsilon)} \cdot 2^{n_c \cdot (1/15 - \epsilon)} \cdot 2^{n_c \cdot (1/15 - \epsilon)} = 2^{n_c (1 - 6\epsilon)}$$

possible collections, *i.e.*, the cardinality of $\mathbf{s}[i+2]$, $\mathbf{s}_0[i]$, and $\mathbf{s}_0[i+1]$. Thus the expected number of incorrect collections which nonetheless have a bin number matching $\mathbf{x}[i+2]$ is

$$2^{-n_c} \cdot 2^{n_c(1-6\epsilon)} = 2^{-6 n_c \epsilon}.$$

Thus the probability of decoding failure goes to zero as $n_c \to \infty$.

Similarly, when $\mathbf{x}[i+3]$ is received the decoder searches for a collection of source components which has a bin number matching $\mathbf{x}[i+3]$. Provided that $\mathbf{s}_0[i+1]$ and $\mathbf{s}[i+2]$ were correctly determined in the preceding step, the probability that an incorrect collection has this bin number will be small. Specifically, the unknown components in $\mathbf{x}[i+3]$ are

$$(\mathbf{s}_0[i+3], \mathbf{s}_1[i+3], \mathbf{s}_2[i+3], \mathbf{s}_3[i+3], \mathbf{s}_1[i], \mathbf{s}_2[i]) \qquad (7.30)$$

because $\mathbf{s}_0[i+2]$ and $\mathbf{s}_0[i+1]$ were recovered when $\mathbf{x}[i+2]$ was decoded. Thus, according to the previous argument $\mathbf{x}[i+3]$ is decoded.

Finally, having decoded $\mathbf{s}_0[i]$, $\mathbf{s}_1[i]$, and $\mathbf{s}_2[i]$ the receiver can decoded $\mathbf{s}_3[i]$ by searching for a sequence $\mathbf{s}_3[i]$ jointly typical with $\mathbf{y}[i]$ and the previously decoded terms. There are

$2^{n_c(2/3-\epsilon)}$ possible values for $\mathsf{s}_3[i]$. The codeword corresponding to the correct value of $\mathsf{s}_3[i]$ will be jointly typical with high probability. The probability that a given (incorrect) value of $\mathsf{s}_3[i]$ is jointly typical with the channel output $\mathsf{y}[i]$ is roughly $2^{-n_c I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})}$. Thus the expected number of incorrect codewords which are jointly typical with $\mathsf{y}[i]$ is roughly

$$2^{n_c[I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})-\epsilon-I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})]} = 2^{-n_c\epsilon}.$$

Thus, with probability arbitrarily close to 1, only the transmitted sequence $\mathsf{s}_3[i]$ will be jointly typical and decoding will succeed.

This examples illustrates an interesting (and probably essential) feature of our encoding and decoding process: only a subset of the information from $\mathsf{s}[i]$ is repeated in later packets. When a burst occurs at time $i$, the decoder does not try to recover all of $\mathsf{x}[i]$ from later packets. Instead, the decoder recovers just enough of $\mathsf{x}[i]$ from later packets to decrease the uncertainty about $\mathsf{x}[i]$ to a point where it can be resolved from the output of the bad channel.

## ■ 7.7 Practical Coding Schemes

The random coding schemes presented so far are too complex to be implemented, but the random codes suggest how practical, structured codes can be designed. As illustrated by the examples in Section 7.6.1 and Section 7.6.2, our coding scheme has a component that is based only on the decoding delay and burst length and a component based only upon the channel mutual informations in the good and bad state. The former corresponds to interleaving and binning scheme while the latter corresponds to using familiar random codes for a noisy channel. It seems that these components are almost separable.

Based on recent success in practical constructions of single path burst erasure correcting codes [119] via Reed-Solomon codes, we suspect this insight has some interesting practical implications. Specifically, we believe that burst-delay capacity achieving codes for non-erasure channels can be designed by combining inner codes for traditional non-bursty channels with outer codes designed for a burst erasure channel. Decoding would work by first using the inner decoder on packets arriving over the good channel and treating packets arriving over the bad channel as erased. Next components 0 through $\mathtt{L}-2$ of the "erased" information sources would be decoded. Finally, all previously decoded information could be used as side information for the inner decoder used to recover packets arriving over the bad channel.

For example, consider a bursty, additive white Gaussian noise channel. One appropriate inner code would be a multi-level Gaussian code where one level (or set of levels) conveys about $I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})$ bits per sample and the other level (or set of levels) conveys about $I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}})$ bits per channel. The outer code would be a burst-delay capacity achieving code for an erasure channel with the same $\mathtt{T}$ and $\mathtt{B}$. First, all source components for $\mathsf{s}_j[i]$ except $\mathsf{s}_{\mathtt{L}-1}[i]$ would be decoded from the outer erasure correcting code. The resulting information would correspond to the second set of levels for the multi-level Gaussian code. "Subtracting off" this known information from the multi-level Gaussian code, would then leave a Gaussian code conveying about $I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})$ bits per sample of information which could be successfully decoded on the bad channel.

## ■ 7.8 Concluding Remarks

In this chapter we have considered streaming information over bursty channels. We introduced an adversarial channel model where nominally each packet is received over a good channel while an occasional burst of B packets are received over a bad channel. One advantage of this adversarial model is that it requires a less precise channel model and hence may be more robust to the difficulties of accurate channel measurement. More importantly, however, our adversarial burst model provides a nice setting to study and illustrate the relationship between channel dynamics and code structure.

Specifically, if we require that each source packet can be decoded with a delay of at most T packets, then a natural quantity of interest is the maximum amount of information which can be reliably communicated. In order to derive a coding theorem characterizing this burst-delay capacity, we introduced a new type of random coding argument which spreads information over time in a way which depends on the burst length and decoding delay. We believe that this type of random coding argument may be an essential tool in understanding the fundamental delay introduced in a variety of channel models both bursty and otherwise.

Chapter 8

# Delay-Optimal Burst Erasure Code Constructions

In this chapter we describe how to construct codes to meet the optimal burst-delay trade-off in Chapter 7. We focus mainly on burst erasure channels, but such codes can also be extended to other types of bursty channels as sketched in Section 8.5.

## 8.1 Problem Model

We consider the same problem model as in Chapter 7, but specialize it to erasure channels and feedback-free linear codes with finite memory. As illustrated in Fig. 8-1 a linear, feedback-free, rate $R = k/n$, $\mathtt{M}$ path packet encoder, $f(\cdot)$, is a causal, multiple-input multiple-output, linear filter (*i.e.*, a convolutional code). Specifically, the input to $f(\cdot)$ is a source stream, $\mathsf{s}[i]$, where each source symbol consists of $k$ sub-symbols $\mathsf{s}[i] = (\mathsf{s}_0[i], \mathsf{s}_1[i], \ldots, \mathsf{s}_{k-1}[i])$ each taking values over some finite field $\mathrm{GF}(q)$. The output of $f(\cdot)$ is a collection of encoded streams, $\mathbf{x}[i,p]$ (with $p \in \{0, 1, \ldots, \mathtt{M}-1\}$) where each encoded symbol consists of $n$ sub-symbols over the same finite field $\mathrm{GF}(q)$. At time $i$, each $\mathbf{x}[i,p]$ is transmitted along the $p$th channel path. We consider bursts of length $\mathtt{B}$ where at any time, $i^*$, $\mathtt{B}$ packets on any path, $p^*$, may be erased. A burst-delay optimal code is one which can correct any such burst with delay $\mathtt{T}$ where the parameters $\mathtt{B}$, $\mathtt{T}$, and $R$ satisfy the trade-off derived in (7.10).

In this chapter, our goal is to construct such optimal codes with low encoding and decoding complexity and the smallest possible field size. In practice, the maximum possible packet size is determined by the underlying channel characteristics and is not a parameter which the code designer can change. For example, the coherence time in a wireless channel or the Maximum Transmit Unit (MTU) in an Internet link might determine the maximum packet size (and hence the maximum field size for the code). Thus if a burst-delay optimal code with an alphabet size smaller than the maximum packet size exists, then the burst-delay capacity in (7.10) can be practically achieved. Otherwise, finite length effects force the system to operate at a burst-delay-rate trade-off worse than the fundamental limit in (7.10).

## 8.2 Single-Link Codes

We begin by focusing on single link codes and observe that designing packet encoders can be viewed as essentially designing matched interleavers and block codes. Specifically, in Fig. 8-1, the initial delay elements can be viewed as a periodic interleaving of the input similar to [62]. Instead of working with the structure in Fig. 8-1 which requires us to specify $\mathtt{M}$ block codes, we instead choose a different interleaving structure to simplify design of the corresponding block code.

In particular, to obtain a simple mapping between the properties of the convolutional code and the block code, we use the interleaving structure shown in Fig. 8-2. We refer to this as “diagonal interleaving” both to distinguish it from Forney’s periodic interleaving [62] and for another reason illustrated shortly. Furthermore, when discussing block code design, we denote the inputs and outputs of the block code as $\mathsf{v}_j$ and $\mathsf{w}_j$ and reserve the symbols $\mathsf{s}_j[i]$ and $\mathsf{x}_j[i]$ for the input and output of the corresponding packet encoder obtained via interleaving.

As stated in the following theorem and proved in Appendix F, diagonal interleaving allows us to construct a rate $k/n$ *convolutional* code to correct any burst of length $B$ with delay $T$, by focusing on the somewhat simpler problem of constructing a single $(n,k)$ *block* code.

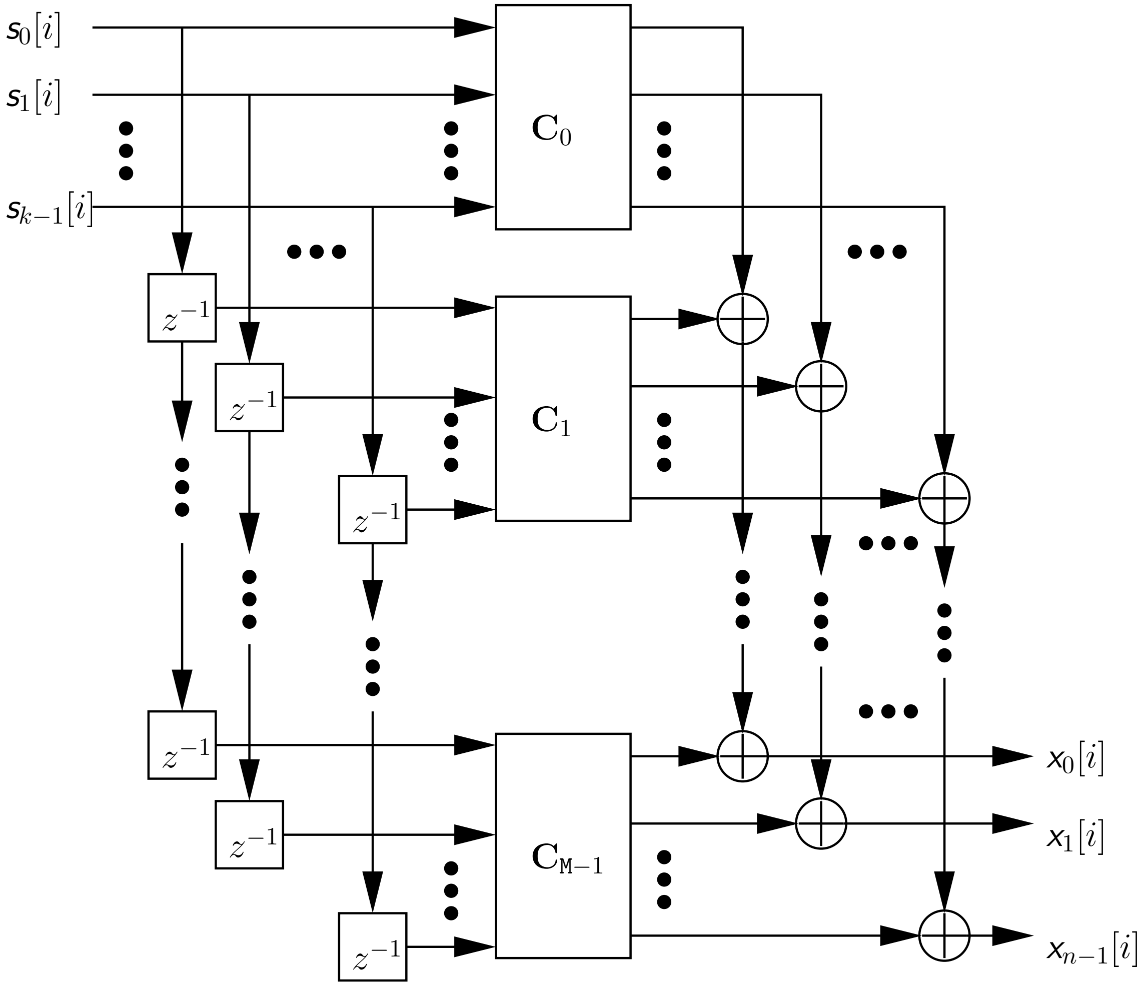


**Figure 8-1.** A linear, feedback-free, rate $k/n$, memory M packet encoder can be represented as a multiple-input multiple-output finite impulse response linear filter (*i.e.*, a convolutional code). For multiple links (*i.e.*, $\mathtt{M} > 1$), a similar section is required for each link. The source packet at time $i$, $\mathbf{s}[i]$, is split into $k$ sub-symbols each taking values in a finite field GF($q$). The sub-symbols are delayed, fed into the sub-codes $\mathbf{C}_j$, and added together to produce the encode packet at time $i$, $\mathbf{x}[i]$. Essentially, the initial delay section can be as interleaving the input before applying a block code.

**Theorem 24.** *Consider an $(n, k)$ systematic block code,* $\mathbf{C}$*, that can recover the ith input symbol from the unerased symbols from 0 through* $\mathtt{T} + i$ *for any erasure burst of length* $\mathtt{B}$*. Then such a code can be diagonally interleaved as in Fig. 8-2 to yield a rate k/n, single-link, packet encoder which can correct any erasure burst of length* $\mathtt{B}$ *with delay* $\mathtt{T}$*.*

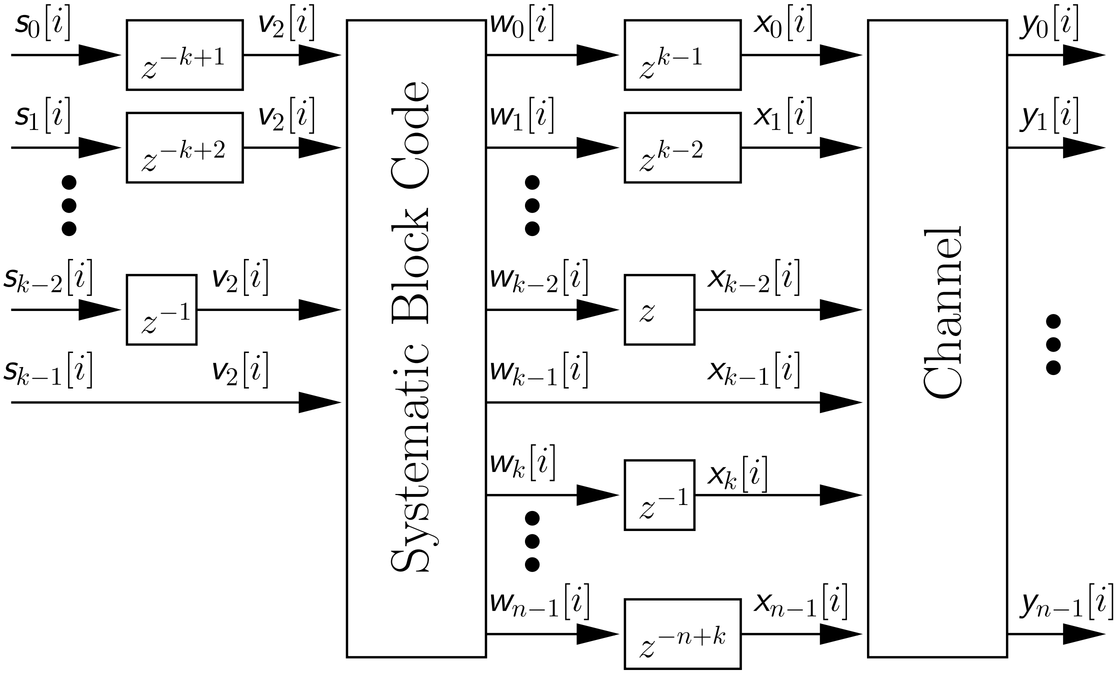


**Figure 8-2.** A convolutional code structure based on diagonal interleaving. The $z^{-\lambda}$ and $z^{\lambda}$ elements denote delay or advance by $\lambda$. Advances are possible since they can be "pushed through" the systematic block code to cancel with delays.

An example best illustrates this idea. Imagine we want to construct a convolutional code to correct all bursts of length $B = 2$ with delay $T = 3$. First, we construct the following block code taking the input vector $\mathbf{v}$ to the output vector $\mathbf{w}$ via left multiplication by a code matrix $\mathbf{C}$:

$$\mathbf{w} = \mathbf{v} \cdot \mathbf{C} = \begin{pmatrix} v_0 & v_1 & v_2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} v_0 & v_1 & v_2 & v_0 + v_2 & v_1 + v_2 \end{pmatrix} \tag{8.1}$$

It is straight-forward to verify that if any two symbols in the code vector $\mathbf{w}$ are erased, then they can both be recovered with a delay of 3. For example if $w_1$ and $w_2$ are erased then the latter can be recovered via $w_2 = w_3 \oplus w_0$ and the former can be recovered via $w_1 = w_4 \oplus w_2$.

Of course the block code in (8.1) does not satisfy our definition of a packet encoder (it can only encode a single block and not a semi-infinite stream as required by our model). Therefore, to obtain a packet encoder, we interleave such a block code as in Fig. 8-2. The effect of the interleaver is that parity check symbols are computed along "diagonals" of the convolutional code as illustrated in Table 8.1. It is easy to show that if the constitunent block code corrects any burst of length $B$ with delay $T$ then the corresponding convolutional code will also. Hence, we can simplify the convolutional code construction problem by only

Table 8.1. Illustration of how a rate 3/5 convolutinal code can be constructed by the interleaving structure of Fig. 8-2. Essentially, the block code is computed along the "diagonal" of the convolutional code. Boxed entries compromise one such constituent block code.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\mathsf{x}[i] =$ | ( | $\boxed{s_0[i]}$ | $s_1[i]$ | $s_2[i]$ | $s_0[i-3] \oplus s_2[i-1]$ | $s_1[i-3] \oplus s_2[i-2]$ | ) |
| $\mathsf{x}[i+1] =$ | ( | $s_0[i+1]$ | $\boxed{s_1[i+1]}$ | $s_2[i+1]$ | $s_0[i-2] \oplus s_2[i]$ | $s_1[i-2] \oplus s_2[i-1]$ | ) |
| $\mathsf{x}[i+2] =$ | ( | $s_0[i+2]$ | $s_1[i+2]$ | $\boxed{s_2[i+2]}$ | $s_0[i-1] \oplus s_2[i+1]$ | $s_1[i-1] \oplus s_2[i]$ | ) |
| $\mathsf{x}[i+3] =$ | ( | $s_0[i+3]$ | $s_1[i+3]$ | $s_2[i+3]$ | $\boxed{s_0[i] \oplus s_2[i+2]}$ | $s_1[i] \oplus s_2[i+1]$ | ) |
| $\mathsf{x}[i+4] =$ | ( | $s_0[i+4]$ | $s_1[i+4]$ | $s_2[i+4]$ | $s_0[i+1] \oplus s_2[i+3]$ | $\boxed{s_1[i+1] \oplus s_2[i+2]}$ | ) |

considering block code constructions.

## 8.2.1 Reed-Solomon Codes Are Not Generally Optimal

Reed-Solomon codes (and other cyclic codes) are well-known to have excellent burst correcting properties when their delay is not considered. Thus, it might seem natural to use such codes in the periodic interleaving structure as suggested by Forney [62] or in the diagonal interleaving structure in Fig. 8-2. Unfortunately, such constructions are not generally optimal. For example, consider attempting to use a Reed-Solomon code in place of the code of (8.1) to correct bursts of length 2 with delay 3. If the delay starts at time 0 and erases $w_0$ and $w_1$, then these symbols can not be recovered until all of them remaining code symbols are received. Thus the Reed-Solomon construction requires a worst case delay of 4 instead of 3 with either type of interleaving.

In general, the straightforward construction using an $(n, k)$ Reed-Solomon code requires a worse case delay of $n-1$ to correct any burst of length $n-k$. In contrast, an $(n, k)$ block code must correct any burst of length $n-k$ with delay $k$ if it is to yield a burst-delay optimal code after interleaving.

## 8.2.2 Provably Optimal Construction For All Rates

As stated Theorem 25 and illustrated in Fig. 8-3, we can construct the required $(n, k)$ block code to interleave into a delay-optimal rate $k/n$ burst correcting convolutional code by combining a repetition block code and a block code which corrects bursts without regard to delay.

**Theorem 25.** *A rate $k/n$ code capable of correcting any burst of length $n-k$ with delay $k$ can be constructed by diagonally interleaving the matrix*

$$\mathbf{G}^*_{k\times n} = \begin{pmatrix} \mathbf{I}_{n-k\times n-k} & \mathbf{0}_{n-k\times 2k-n} & \mathbf{I}_{n-k\times n-k} \\ \mathbf{0}_{2k-n\times n-k} & \multicolumn{2}{c}{\mathbf{P}_{\text{cyc},2k-n\times k}} \end{pmatrix} \tag{8.2}$$

*where $\mathbf{I}_{i\times j}$ is an i-by-j identity matrix, $\mathbf{0}_{i\times j}$ is an i-by-j zero matrix, and $\mathbf{P}_{\text{cyc},2k-n\times k}$ is the $(2k-n)$-by-$k$ systematic generator matrix for a length $k$, dimension $2k-n$ code capable of correcting any erasure burst of length at most $n-k$ (including end-around bursts) without regard to delay.*

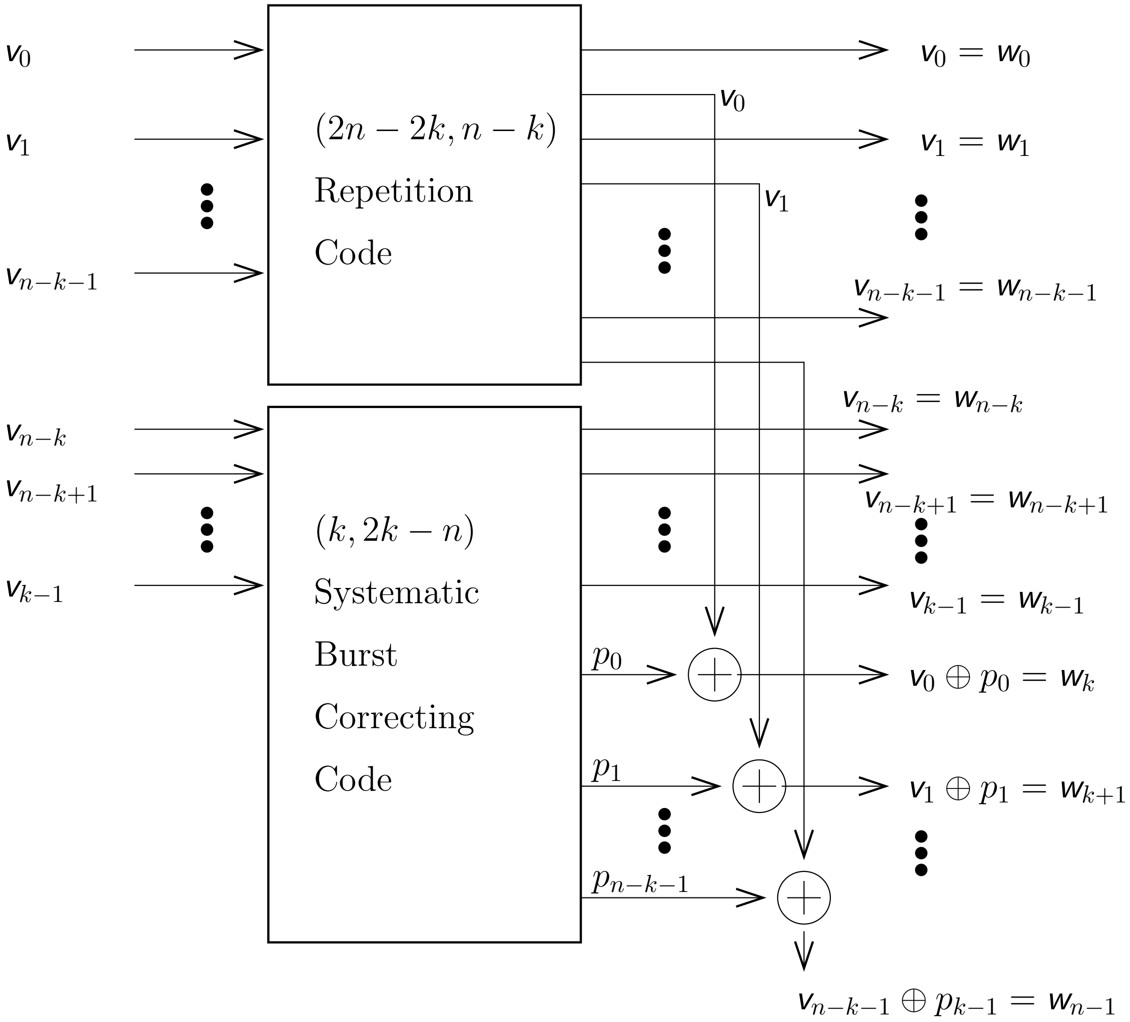


**Figure 8-3.** An $(n,k)$ delay-optimal erasure burst correcting block code can be constructed by combining a repetition code and a code capable of correcting erasure bursts of length $n-k$ without regard to delay. The encoder shown above maps the (interleaved) symbols $v_0$, $v_1$, …, $v_{k-1}$ to the (interleaved) code symbols $w_0$, $w_1$, …, $w_{n-1}$.

Before proving Theorem 25, we illustrate how it can be applied to construct a rate 7/11 code over GF(2) which is capable of correcting any burst of length 4 with delay 7. The desired code requires a generator matrix $\mathbf{P}_{\mathrm{cyc},7\times 3}$ corresponding to a $(7,3)$ erasure burst correcting code. We can obtain $\mathbf{P}_{\mathrm{cyc},7\times 3}$ from a binary cyclic code generated by $g(x) = x^4 + x^2 + x^1 + 1$. Thus the desired generator matrix for the rate 7/11 code is

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{8.3}$$

*Proof of Theorem 25:* As illustrated by Fig. 8-3 and the example code in (8.3), the first $n-k$ information symbols are treated quite differently than the last $2k-n$ information symbols. We refer to these two classes of information symbols as the urgent and non-urgent symbols respectively since we can ignore the delay required in recovering the latter provided they are eventually recovered but must insure the delay in recovering the former is at most $k$. We first describe how the non-urgent symbols are decoded and then describe how the urgent symbols are decoded.

Assume that an erasure burst of length $n-k$ starts at time $0 \leq i \leq k-1$ (bursts starting after time $k-1$ are equivalent to bursts starting at time $k-1$ since the bursts after time $k-1$ fall off the end of the code vector). By time at most $i+k-1$, we would receive exactly $2k-n$ cyclically consecutive[1] symbols of the bottom block code in Fig. 8-3. Note that while repeated symbols from time 0 through $i-1$ are added to the parity check symbols of the bottom block code, these repeated symbols can be subtracted out of the desired parity check symbols of the bottom block code because they are correctly received before the burst. By assumption the bottom block code can decode any cyclic burst of length $n-k$ or equivalently any cyclically consecutive $2k-n$ symbols are an information set. Thus by time $i+k-1$ we can recover the non-urgent information symbols.

Once the non-urgent information symbols are recovered we can subtract the effect of the parity check symbols of the bottom block code in Fig. 8-3 from the top repetition code. Furthermore, note that this subtraction can be performed at time $i+k-1$ which is exactly the time when the first urgent symbols must be recovered. Thus the first urgent information symbol is recovered from the repetition code at time $i+k-1$, the second urgent information symbol is recovered from the repetition code at time $i+k$, and so on until the last urgent information symbol is recovered.

For example, consider the matrix in (8.3). If a burst started at time 2 then urgent information symbols 0 and 1 would be received correctly as would the last non-urgent information symbol. The 2 urgent information symbols could then be subtracted out of the first 2 parity check symbols of the bottom block code. After this subtraction is performed, at time 9 we have 3 consecutive symbols of the bottom $(7,3)$ code and thus the non-urgent information symbols are recovered. Once the non-urgent information symbols are recovered, their effect on the repetition code at the top of Fig. 8-3 can be subtracted out. Thus information symbols 2, 3, and 4 would each be recovered with delay 7. □

[1] By cyclically consecutive we mean that the code vector is assumed to wrap around. For example, symbols $A$—$D$ in the code vector $(A\ B\ *\ *\ C\ D)$ are cyclically consecutive.

At first glance, Theorem 25 may seem useless since applying it requires constructing a smaller burst correcting code. Since the smaller burst correcting code need only correct bursts without regard to delay, however, constructing it is generally much easier than directly constructing a low delay burst correcting code. In fact, if a cyclic code of the appropriate length and dimension exists for the desired alphabet then it suffices since any $(n,k)$ cyclic code can correct all erasure bursts of length $n-k$.[2]

For example, to construct a rate 5/8 code capable of correcting any burst of length 3 with delay 5 we require the generator matrix for a $(5,2)$ code which can correct any burst of length 3 (without regard for delay). Since $x^5+1$ factors into $(1+x)\cdot(1+x+x^2+x^3+x^4)$ there is no binary cyclic code of dimension 2 and length 5.[3] We could consider larger fields to find a cyclic code with the right parameters (*e.g.*, a Reed-Solomon code over GF(5)). If using a binary alphabet is important, we can use a (non-cyclic) binary code generated by the following matrix:

$$\mathbf{P}_{\mathrm{cyc},2\times 5} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \tag{8.4}$$

which we can verify by inspection can correct any erasure burst of length 3. Thus we obtain the generator matrix for the desired rate 5/8 code

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}. \tag{8.5}$$

In any case, if a cyclic code does not exist or is inconvenient to find, the required erasure burst correcting code can always be constructed with a symbol alphabet that is not too large.

**Corollary 7.** *The largest alphabet size required for the generator matrix* $\mathbf{P}_{\mathrm{cyc},2k-n\times k}$ *in Theorem 25 is at most* $q^m$ *where* $q^m$ *is the smallest prime power exceeding* $k$.

*Proof.* By constructing a Reed-Solomon code over GF($q^m$) and puncturing the code down to length $k$ we obtain an MDS code which can correct the desired erasure bursts. □

**Corollary 8.** *The largest alphabet size required for the generator matrix* $\mathbf{P}_{\mathrm{cyc},2k-n\times k}$ *in Theorem 25 is at most* $2k$.

*Proof.* By applying Corollary 7 with $q=2$ we see that the field size is at most the largest power of 2 which is greater than or equal to $k$. □

While there are clearly parameters for which no cyclic code exists, it would be interesting to know whether there are parameters $(n,k)$ where there is no binary block code capable of correcting all erasure bursts of length $(n-k)$ and moving to higher alphabets is absolutely required.

[2] To see this, note that for a systematic cyclic code the first $k$ symbols form an information set. Since any cyclic shift of a codeword is also a codeword, any consecutive $k$ symbols must form an information set.

[3] A cyclic code of dimension $k$ and length $n$ over GF(2) exists if and only if $x^n+1$ has a factor of degree $n-k$ over GF(2) [108].

## ■ 8.3 Two-Link Codes

In this section we consider scenarios where the transmitter can use two links as illustrated in Fig. 8-4. Intuitively, the advantage of using multiple links is that the extra links can reduce the delay required to correct for a given burst length and coding rate. In particular, it is possible to lower bound the ratio of decoding delay to correctable burst length via

$$\frac{\mathtt{T}}{\mathtt{B}} \geq \begin{cases} \max\left[\frac{1}{\mathtt{M}} \cdot \frac{1}{1-\mathtt{R}} - 1, 1\right], & \mathtt{R} \geq \frac{\mathtt{M}-1}{\mathtt{M}} \\ 0, & \mathtt{R} \leq \frac{\mathtt{M}-1}{\mathtt{M}} \end{cases} \tag{8.6}$$

or equivalently to upper bound the code rate via

$$\mathtt{R} \leq \begin{cases} 1 - \frac{\mathtt{B}}{(\mathtt{T}+\mathtt{B})\cdot\mathtt{M}}, & \mathtt{T} \geq \mathtt{B} \\ 1 - \frac{1}{\mathtt{M}}, & \mathtt{T} < \mathtt{B}. \end{cases} \tag{8.7}$$

As we see from (8.6) doubling the number of transmission links almost halves the decoding delay confirming our intuition about the value of multiple links.

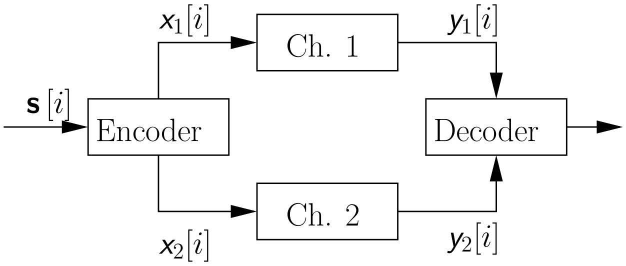


**Figure 8-4.** A channel with two transmission links.

## ■ 8.3.1 Constructions

To construct delay optimal *convolutional* codes for the two-link case we again use the idea of interleaving *block* codes as illustrated in Table 8.2. To construct a delay optimal convolutional code we can instead solve the simpler problem of constructing a delay optimal block code. Notice that in contrast to the single-link case, however, the block code for a multi-link channel has multiple samples for a given "time slot". Perhaps a good way to denote such a block code is to write the output for link $j$ as $\mathbf{w}_j$ and represent the encoding operation as

$$\left( \mathbf{w}_1 \,\|\, \mathbf{w}_2 \right) = \mathbf{v} \cdot \left( \mathbf{C}_1 \,\|\, \mathbf{C}_2 \right) = \left( \mathbf{v} \cdot \mathbf{C}_1 \,\|\, \mathbf{v} \cdot \mathbf{C}_2 \right) \tag{8.8}$$

where the $||$ is intended to convey the idea that $\mathbf{w}_1$ and $\mathbf{w}_2$ are code vectors which are sent at the same time but on different links.

With this notation, the constituent block code in Table 8.2 would have

$$\mathbf{C}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \text{ and } \mathbf{C}_2 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \text{ with } \mathbf{v} = \begin{pmatrix} v_0 & v_1 & v_2 \end{pmatrix} \tag{8.9}$$

**Table 8.2.** Illustration of how a two-link convolutional code can be constructed by using a block code on the diagonal. Boxed entries compromise a constituent block code. The code shown is a rate 3/4 code capable of correcting any burst of length $\mathtt{B} = 1$ with decoding delay $\mathtt{T} = 1$ and is therefore optimal according to (8.6).

$$\begin{array}{rl}
\mathsf{y}_0[i] = & \left( \boxed{\mathsf{s}_0[i]} \quad \mathsf{s}_1[i] \right) \Big\| \left( \boxed{\mathsf{s}_2[i]} \quad \mathsf{p}_1[i] \right) \\
\mathsf{y}_1[i] = & \left( \mathsf{s}_0[i+1] \quad \boxed{\mathsf{s}_1[i+1]} \right) \Big\| \left( \mathsf{s}_2[i+1] \quad \boxed{\mathsf{p}_1[i+1] = \mathsf{s}_0[i] \oplus \mathsf{s}_1[i+1] \oplus \mathsf{s}_2[i]} \right)
\end{array}$$

resulting in the two-link block code

$$\left( \mathbf{w}_1 \, \| \, \mathbf{w}_2 \right) = \left( \mathbf{v} \cdot \mathbf{C}_1 \, \| \, \mathbf{v} \cdot \mathbf{C}_2 \right) = \left( \begin{array}{c|c} \mathsf{v}_0 & \mathsf{v}_2 \\ \mathsf{v}_1 & \mathsf{v}_0 \oplus \mathsf{v}_1 \oplus \mathsf{v}_2 \end{array} \right). \tag{8.10}$$

Fig. 8-5 illustrates how our code construction can be extended to the case of two links. As stated Theorem 26 (and proved in Appendix F), we can obtain a block code to interleave into a two-link, delay-optimal, rate $k/n$, burst correcting convolutional code by combining smaller delay-optimal single-link codes.

**Theorem 26.** *Consider positive integers $k$ and $n$ with $n$ even and $k/n \geq 3/4$. A rate $k/n$ two-link code capable of correcting any burst of length $\mathtt{B} = (n-k)$ with delay $\mathtt{T} = (k - n/2)$ can be constructed by diagonally interleaving the matrix $\mathbf{G}^{**}_{k \times n} = (\mathbf{G}^{(**,1)}_{k \times n} || \mathbf{G}^{(**,2)}_{k \times n})$ where*

$$\mathbf{G}^{(**,1)}_{k \times n} = \begin{pmatrix} \mathbf{I}_{n/2 \times n/2} \\ \mathbf{0}_{(k-n/2) \times n/2} \end{pmatrix}, \quad \mathbf{G}^{(**,2)}_{k \times n} = \begin{pmatrix} \mathbf{G}^{(*,0)}_{(k-n/2) \times n/2} \\ \mathbf{0}_{(n-k) \times (k-n/2)} \quad \mathbf{I}_{(n-k) \times (n-k)} \\ \mathbf{G}^{*}_{(k-n/2) \times n/2} \end{pmatrix}, \tag{8.11}$$

*and $\mathbf{G}^{(*,0)}_{a \times b}$ is a generator for the single-link code construction in Theorem 25 with the upper left identity matrix replaced by zeros, and the remaining notation the same as in Theorem 25.*

Some discussion and examples help clarify Theorem 26. First, note that restricting $n$ to be even is only used to simplify the exposition and does not restrict the possible rates. For example, to obtain a rate 4/5 code, one could use the parameters $k = 8$ and $n = 10$ in Theorem 26. Second, the restriction that $k/n \geq 3/4$ causes no difficulty in achieving the bound in (8.6). To see this note that for rates less than 3/4, the two-link bound yields $\mathtt{T}/\mathtt{B} \geq 0$ which is trivially achieved by sending exactly the same data on both links. Evidently, only codes with $\mathtt{R} = 1/2$ or $\mathtt{R} \geq 3/4$ are sufficient. Finally, observe that by setting $\mathtt{R} = k/n$ and $\mathtt{M} = 2$ in (8.6) we obtain $\mathtt{T}/\mathtt{B} \geq (k - n/2)/(n-k)$ so the codes in Theorem 26 are optimal.

To gain some insight into the structure of the two-link construction we consider a rate 4/5 example. For this rate, we choose $n = 10$ and $k = 8$ to obtain a code capable of correcting any burst of length 2 with delay 3. To obtain codes correcting longer bursts, we could multiply this $k$ and $n$ by any positive integer.

According to Theorem 26, this code requires the matrix $\mathbf{G}^{*}_{3 \times 5}$ corresponding to a $(5,3)$ single-link, delay-optimal, erasure burst correcting block code. Using Theorem 25 we obtain

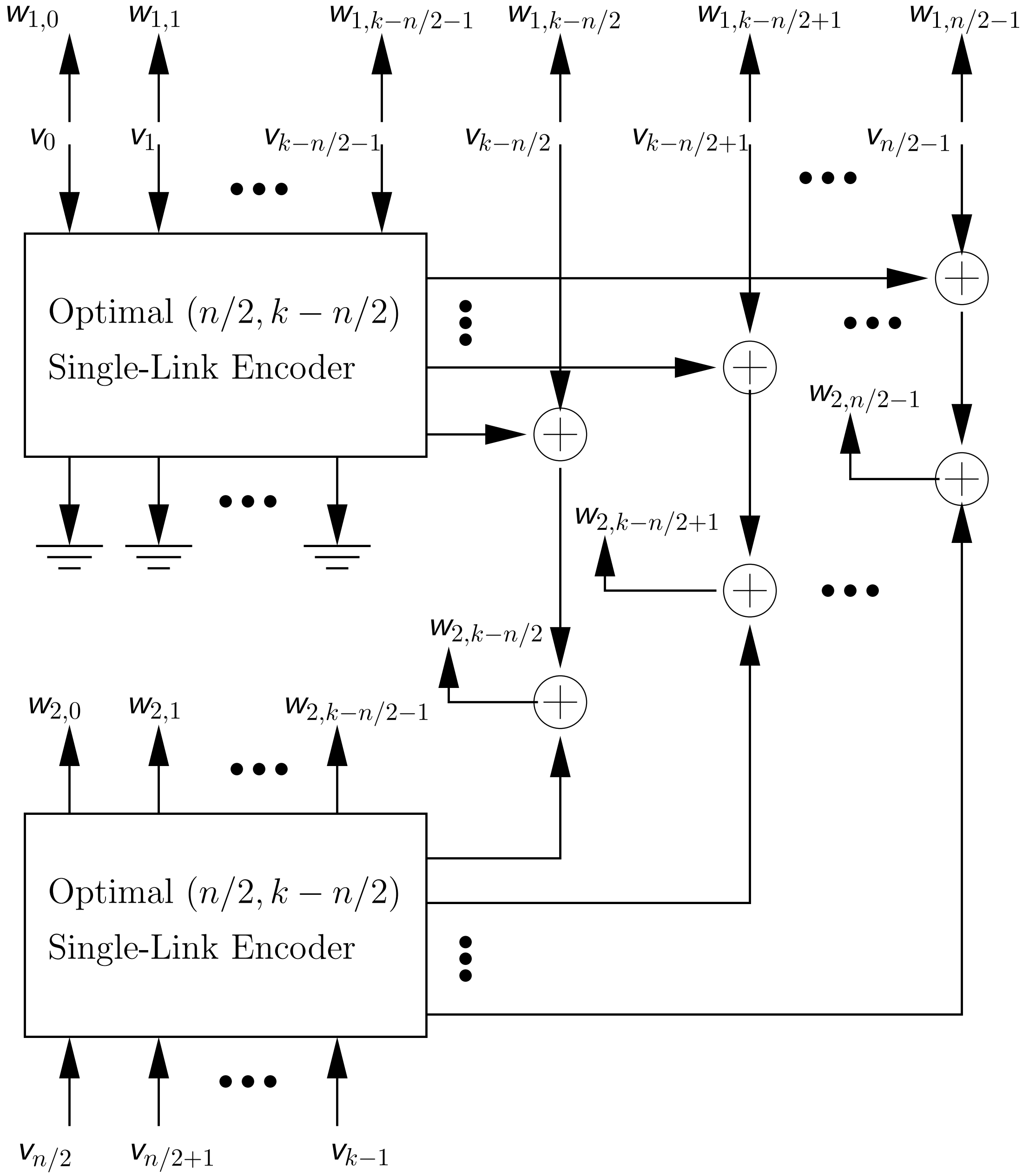


**Figure 8-5.** An $(n, k)$ two-link, delay-optimal, erasure burst correcting, block code correcting any burst of length $n-k$ with delay $k-n/2$ can be constructed by combining two single-link, $(k-n/2, n/2)$ delay-optimal, erasure burst correcting, block codes. The encoder shown above maps the $k$ (interleaved) symbols $v_0$, $v_1$, …, $v_{k-1}$ to the $n$ (interleaved) code symbols $w_{1,0}$, $w_{1,1}$, …, $w_{1,n/2-1}$ and $w_{2,0}$, $w_{2,1}$, …, $w_{2,n/2-1}$. The systematic output symbols of each single-link code are shown vertically and the parity output symbols are shown horizontally. Note that the first code stream $w_{1,0}$, $w_{1,1}$, …, $w_{1,n/2-1}$ is simply composed of the first $n/2-1$ input symbols and so the systematic output symbols of the top single-link code are unused.

$$\mathbf{G}^{*}_{3\times 5} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \tag{8.12}$$

Thus the encoding matrix for the two-link block code is

$$\mathbf{G}^{**}_{8\times 10} = \left(\begin{array}{ccccc||ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right). \tag{8.13}$$

Let us consider the burst correcting capabilities of $\mathbf{G}^{**}_{8\times 10}$. As illustrated by Fig. 8-5 and the encoding matrix in (8.13), the idea behind this construction is that if a burst occurs on the second stream, we strip out the information received on the first stream so that the second stream looks like a single-link code suffering a correctable burst. A similar stripping argument can be used if a burst occurs on the first stream. This intuition is made precise by considering bursts on each stream in detail as follows.

Imagine an erasure burst of length 2 occurs on the second stream. Without loss of generality, we can assume that the data symbols on the first stream ($v_0$, $v_1$, ..., $v_4$) are zero. If this is not the case, they can be subtracted out since they are received correctly. If the first stream can be ignored, then we can ignore the first 5 rows of the encoding matrix and collapse it to

$$\left(\begin{array}{ccccc||ccccc} \cancel{1} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} \\ \cancel{0} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} \\ \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{1} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right). \tag{8.14}$$

If we ignore the left half of the resulting matrix since it is all zeros, we are left with the encoding matrix for a single-link code which can correct any erasure burst of length 2 with delay 3. Thus we can correct the second stream with delay 3.

Now imagine that instead an erasure burst of length 2 occurs on the first stream. The data on the second stream is received without error and can therefore assumed to be zero or subtracted off. If the erasure burst on the first stream does not affect the last two symbols on the first stream, then these can be stripped off and we are left with the effective encoding matrix

$$\left(\begin{array}{ccccc||ccccc}
1 & 0 & 0 & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & 1 & 0 \\
0 & 1 & 0 & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & 0 & 1 \\
0 & 0 & 1 & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & 1 & 1 \\
\cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} \\
\cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} \\
\cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} \\
\cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{1} \\
\cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{1} & \cancel{1}
\end{array}\right). \tag{8.15}$$

Recalling Fig. 8-5 may help clarify this process. Specifically, the parity check symbols resulting from encoding the first 3 symbols on the first stream were added to the parity check symbols on the second stream as well as the last two symbols of the first stream. Thus if the latter are known, the effective encoding matrix in (8.15) is exactly the encoding matrix for a single-link code capable of correcting any burst of length 2 with delay 3. So the remaining data symbols can be recovered as described in Theorem 25.

Next imagine that an erasure burst of length 2 affects either of the last two symbols on the first stream. In this case it might appear that our goal of making the first stream look like a single-link code fails because we can no longer subtract the erased symbols on the first stream from the parity check symbols on the second stream. To be more specific, imagine that an erasure burst of length 2 erases $w_{1,2} = v_2$ and $w_{1,3} = v_3$. Since $v_3$ is unknown it can not be subtracted from $w_{2,3}$.

Thus even though it is $v_3$ that is actually erased, this has the effect of "virtually" erasing $w_{2,3}$. So $v_0$ and $v_1$ are received correctly, $v_2$, is actually erased, $w_{2,3}$ is virtually erased and $w_{2,4}$ is received correctly. But this looks just like an erasure burst of length 2 on a single-link code and can therefore be corrected within the required delay constraint. Essentially, the virtual erasures do not hinder decoding because the top single-link code in Fig. 8-5 can correct them. Finally, once $v_0$, $v_1$, and $v_2$ are recovered, $v_3$ can be recovered from $w_{2,3}$.

## ■ 8.4 Multi-Link Codes

The same ideas used in generalizing single-link codes to make two-link codes can be applied to general multi-link channels. For example, the matrix for a three-link block code

$$\mathbf{G}^{***}_{k\times n} = \left( \mathbf{G}^{(***,1)}_{k\times n} \,\middle\|\, \mathbf{G}^{(***,2)}_{k\times n} \,\middle\|\, \mathbf{G}^{(***,3)}_{k\times n} \right) \tag{8.16a}$$

can be created by choosing

$$\mathbf{G}^{(***,1)}_{k\times n} = \begin{pmatrix} \mathbf{I}_{n/3\times n/3} \\ \mathbf{0}_{(k-n/3)\times n/3} \\ \mathbf{0}_{(k-2n/3)\times n/3} \end{pmatrix}, \tag{8.16b}$$

$$\mathbf{G}^{(***,2)}_{k\times n} = \begin{pmatrix} \mathbf{0}_{(n/3)\times n/3} \\ \mathbf{I}_{n/3\times n/3} \\ \mathbf{0}_{(k-2n/3)\times n/3} \end{pmatrix}, \tag{8.16c}$$

and

$$\mathbf{G}_{k\times n}^{(***,3)} = \begin{pmatrix} \mathbf{G}^{(*,0)}_{(k-2n/3)\times n/3} \\ \mathbf{0}_{(n-k)\times(k-2n/3)} \quad \mathbf{I}_{(n-k)\times(n-k)} \\ \mathbf{G}^{(*,0)}_{(k-2n/3)\times n/3} \\ \mathbf{0}_{(n-k)\times(k-2n/3)} \quad \mathbf{I}_{(n-k)\times(n-k)} \\ \mathbf{G}^{*}_{(k-2n/3)\times n/3} \end{pmatrix}. \tag{8.16d}$$

The version of Fig. 8-5 appropriate for a three-link code would have two stages like the top of Fig. 8-5 and one stage like the bottom. Specifically, the first and second coded streams would simply contain input data. The third coded stream would be produced by feeding $k-2n/3$ input symbols into a single-link code, adding the resulting parity check symbols to the parity check symbols resulting from passing the first $k-2n/3$ input symbols from the other two streams into a single-link code and finally adding the remaining input symbols from the first two streams.

The version of Fig. 8-5 appropriate for an $\mathtt{M}$-link code would have $\mathtt{M}-1$ stages like the top of Fig. 8-5 and one stage like the bottom. Specifically, the first $\mathtt{M}-1$ coded streams would simply contain $n/3$ symbols of input data. The last coded stream would be produced by feeding the last $k-n\cdot(\mathtt{M}-1)/\mathtt{M}$ input symbols into an optimal $(n/3, k-n\cdot(\mathtt{M}-1)/\mathtt{M})$ single-link code, adding the resulting parity check symbols to the parity check symbols resulting from passing the first $k-n\cdot(\mathtt{M}-1)/\mathtt{M}$ input symbols from each of the other streams through similar single-link codes, and finally adding the remaining input symbols from the first $\mathtt{M}-1$ streams.

As stated in the following theorem and proved in the appendix, an optimal $\mathtt{M}$-link code can be developed by extending these ideas in the straightforward manner.

**Theorem 27.** *Consider positive integers $k$, $n$, and $\mathtt{M}$ with $n$ a multiple of $\mathtt{M}$ and $k/n \geq 1-2/\mathtt{M}$. A rate $k/n$, $\mathtt{M}$-link code capable of correcting any burst of length $\mathtt{B}=(n-k)$ with delay $\mathtt{T}=(k-n+n/\mathtt{M})$ can be constructed by diagonally interleaving the matrix*

$$\mathbf{G}_{k\times n}^{(*^{\mathtt{M}},1)} = (\mathbf{G}_{k\times n}^{(*^{\mathtt{M}},1)}||\mathbf{G}_{k\times n}^{(*^{\mathtt{M}},2)}||\dots||\mathbf{G}_{k\times n}^{(*^{\mathtt{M}},\mathtt{M})}) \tag{8.17}$$

*where each $\mathbf{G}_{k\times n}^{(*^{\mathtt{M}},1)}$ is constructed by extending the patterns for $\mathtt{M}=2$ and $\mathtt{M}=3$ in (8.11) and (8.16).*

## ■ 8.5 Codes For Stochastically Any Degraded Channel

Although we only focused on burst erasure channels, our codes can be extended to achieve the optimal burst-delay-rate trade-off in Chapter 7 for a wide variety of stochastically degraded broadcast channels. Essentially, an inner broadcast code can be used to divide the data for each packet into two messages of rate $R_1$ and $R_2$. The former corresponds to the mutual information for the bad channel state while the latter corresponds to the remaining transmission rate. By construction the message of rate $R_1$ can be decoded regardless of the channel state. The message of rate $R_2$ is effectively erased for all packets received over the bad channel state. Thus, by applying the codes developed in this chapter to the second message, the optimal burst-delay-rate trade-off can be achieved for any stochastically degraded channel.

## ■ 8.6 Concluding Remarks

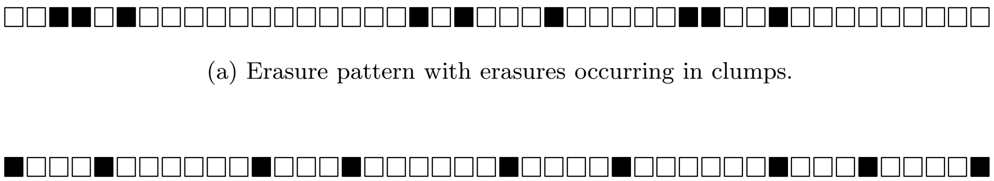
In this chapter we have described how to construct codes with the optimal burst-delay-rate trade-off derived in Chapter 7. We approached the problem by first introducing a diagonal interleaving structure. By using interleaving, we were able to take the general problem which essentially corresponds to designing a convolutional code and reduce it to the simpler problem of designing a block code which can correct erasure bursts with low delay. Furthermore, we reduced the problem of constructing a low delay burst correcting block code to constructing a smaller block code with maximum erasure burst capability without regard to delay. The final code needed in our reduction can always be constructed from Reed-Solomon codes with modest alphabet sizes or sometimes directly to obtain smaller alphabet sizes.

Chapter 9

# Delay Universal Streaming Codes

In contrast to Chapter 7 which considered bursty channels, in this chapter we study the rate-delay trade-off in designing a system for a variety of channel dynamics. Robust communication over different types of channels has been studied with the traditional broadcast channel model [43], the compound channel model [101,187,25] and more recently the static broadcast model [163, 164] and digital fountain codes [31, 32, 33, 114]. Essentially, these models all consider scenarios with different messages, mutual information, or optimal channel input distributions for the different receivers. In contrast, we are interested in channels with different dynamics.

For example, as illustrated by the two erasure channel output sequences in Fig. 9-1, two channels may be able to support the same information rate, but have different dynamics. Specifically, even though the channel outputs in Fig. 9-1(a) and Fig. 9-1(b) have the same number of erasures, these erasures are clumped together in the former and spread out in the latter. Intuitively, these different kinds of channel dynamics may result in different delays or require different coding structures.



(a) Erasure pattern with erasures occurring in clumps.

(b) Erasure pattern with erasures occurring far apart.

**Figure 9-1.** Channels with same number of erasures but different dynamics.

The techniques we consider can be applied to single user settings with channel uncertainty. But, as discussed in the introduction in Chapter 1, perhaps the simplest motivating example is to consider two users receiving the same data over two different channel conditions. We are mainly interested in minimizing the delay for both users simultaneously or, when this is not possible, we are interested in the delay trade-off between the two users.

Specifically, consider transmission of a single message which can be received over a collection of channel ensembles denoted by $\{\mathbf{\Theta}_i\}$ and defined in detail in Section 9.2. The decoding delay will of course depend upon the particular channel conditions encountered. But, codes which minimize delay for $\mathbf{\Theta}$ may be poor for $\mathbf{\Theta}'$. Therefore, achieving good overall performance depends on designing codes which perform well for each $\mathbf{\Theta} \in \{\mathbf{\Theta}_i\}$. For example, in a multicast scenario, user $i$ may receive the signal over channel ensemble $\mathbf{\Theta}_i$. Similarly, even with only a single receiver, there may be some uncertainty about the channel or the channel may act differently at different times and each $\mathbf{\Theta}_i$ may correspond to one such possible channel. Ideally, we would like a system which achieves the minimum possible decoding delay for each possible channel condition.

To illustrate the shortcomings of using traditional block codes, consider a packet loss channel which behaves in one of two possible modes. In the first mode, denoted $\mathbf{\Theta}_1$, no more than 1 packet is lost within a given window of time. In the second mode, $\mathbf{\Theta}_2$, up to 3 packets may be lost in the time window of interest. The former corresponds to Fig. 9-1(b) while the latter corresponds to Fig. 9-1(a).

With a traditional block code, the transmitter could encode each group of 9 source packets, $\mathbf{x}[0], \mathbf{x}[1], \ldots, \mathbf{x}[8]$ into 12 coded packets, $\mathbf{y}[0], \ldots, \mathbf{y}[11]$ using a systematic $(12, 9)$ Reed-Solomon (RS) code (or any other equivalent Maximum Distance Separable code). With this approach, any pattern of 3 (or less) packet losses occurring for channel $\mathbf{\Theta}_2$ can be corrected. But, on channel $\mathbf{\Theta}_1$, if only $\mathbf{y}[0]$ is lost, the soonest it can be recovered is when 9 more coded packets are received. Evidently the system may incur a decoding delay of 9 packets just to correct a single packet loss.

To decrease the delay, instead of using a $(12, 9)$ code, each block of 3 source packets could be encoded into 4 coded packets using a $(4, 3)$ RS code. Since one redundant packet is generated for every three source packets, this approach requires the same redundancy as the $(12, 9)$ code. With the $(4, 3)$ code, however, if the channel is in mode $\mathbf{\Theta}_1$ and only $\mathbf{y}[0]$ is lost, it can be recovered after the remaining 3 packets in the block are received. Thus the $(4, 3)$ system incurs a delay of only 3 to correct one lost packet. While this delay is much smaller than with the $(12, 9)$ code, if more than one packet in a block is lost on channel $\mathbf{\Theta}_2$, then decoding is impossible with the $(4, 3)$ code.

Both practical block codes as well as traditional information theoretic arguments are not designed for real-time systems. Thus we see that minimizing delay for minor losses from $\mathbf{\Theta}_1$ and maximizing robustness for major losses from $\mathbf{\Theta}_2$ are conflicting objectives. Is this trade-off fundamental to the nature of the problem, or is it an artifact of choosing a poor code structure? We show that in many cases of practical interest, there exist better code structures which are universally optimal for all channel conditions. Specifically, for the packet loss example, there exist codes with both the low decoding delay of the $(4, 3)$ code and the robustness of the $(12, 9)$ code.

We begin by considering previous work in Section 9.1 and defining the problem model in Section 9.2. Next we prove a coding theorem characterizing sufficient conditions for the existence of delay universal codes in Section 9.3. In Section 9.4 we consider the packet loss channel and describe a linear code construction which is delay universal and can be encoded and decoded in polynomial time. Using techniques from the theory of random walks, we derive bounds on moments of the decoding delay and evaluate these bounds for the packet loss channel in Section 9.5.

## ■ 9.1 Previous Work

Researchers have explored channel uncertainty and transmission to multiple users by studying the broadcast channel, compound channel, and arbitrarily varying channel, [43] [102]. In these problems, each channel $\mathbf{\Theta}_i$ represents a different channel probability law and the focus is on finding the capacity region. Usually there is a trade-off and increasing the rate received for a particular channel $\mathbf{\Theta}_i$ requires decreasing the rate received over a different channel $\mathbf{\Theta}_{i'}$. In certain cases, however, universal systems exist [103].

In contrast to these static models where the goal is to find various capacity regions, we consider a dynamic model and study the delays or lags. Since we analyze the intra-message delays appropriate for real-time applications as opposed to the overall message delay appropriate for non real-time applications, we require a communication model where small chunks (or packets) of information are generated, sent, received, and decoded. We let each $\mathbf{\Theta}_i$ represent an ensemble of channel conditions, and study the lag between when packets are generated and decoded.

An alternative model also concerned with delays in a somewhat different setting (i.i.d. channels as opposed to the adversarial channels considered here) has been recently studied by Sahai [158]. Also, Shulman and Feder have considered a static broadcast channel where multiple receivers listen for a single long message (*e.g.*, a file transfer) [164]. They derive the trade-off between decreasing total delay for one receiver at the cost of increasing delay for another by studying the delay region. Similarly, Luby *et al.* construct codes to allow transmission of a file over a packet network such as the Internet [114] [33]. Once a number of bits corresponding to the total length of the file have been received, their codes allow the receiver to successfully recover the file with low computational delay. Techniques from this work may also prove useful for real-time applications (especially in the construction of practical codes).

## ■ 9.2 Stream Coding System Model

We consider a streaming model where at each time step $i$ a new source packet $\mathbf{s}[i]$ is revealed to the transmitter and encoded into a channel packet $\mathbf{x}[i]$. Each source packet consists of $n_s$ samples from the alphabet $\mathcal{S}$ and each channel packet consists of $n_c$ samples from the alphabet $\mathcal{X}$. A memory $\mathtt{M}$ packet encoder, $\mathbf{C}$, consists of a causal mapping from the past $\mathtt{M}$ source packets, $\mathbf{s}[i-\mathtt{M}]$, $\mathbf{s}[i-\mathtt{M}+1]$, …, $\mathbf{s}[i-1]$ and the current source packet $\mathbf{s}[i]$ into the current channel input packet $\mathbf{x}[i]$:

$$\mathbf{C} : (\mathcal{S}^{n_s})^{\mathtt{M}+1} \mapsto \mathcal{X}^{n_c}. \tag{9.1}$$

For ease of analysis, we allow $\mathtt{M}$ and $n_c$ to be as large as required. We define the rate of the system as

$$\mathtt{R} = (n_s/n_c)\log|\mathcal{S}|. \tag{9.2}$$

where $|\cdot|$ denotes cardinality of a set. Throughout the chapter log refers to the natural logarithm and thus rate is measured in nats.

To study systems where the channel state changes relatively slowly we use a piece-wise constant or block-interference model [126] [23] [93]. Specifically, the channel output packet, $\mathbf{y}[i]$, obtained by the receiver is determined according to the channel law

$$\prod_{i=0}^{\infty} p_{\mathbf{y}[i]|\mathbf{x}[i];\boldsymbol{\theta}[i]}(\mathbf{y}[i]\,|\mathbf{x}[i]\,;\theta\,[i]) = \prod_{i=0}^{\infty}\prod_{j=1}^{n_c} p_{y|x;\boldsymbol{\theta}}(y_j[i]|x_j[i];\theta\,[i]).$$

where $\boldsymbol{\theta}[i]$ denotes the channel state for packet $i$. In the first part of the chapter, we mostly consider deterministic or adversarial channel state sequences to facilitate analysis and code design. We then compute the performance of the resulting codes for random channel state sequences in the last part of the chapter.

A packet decoder, $\mathbf{C}^{-1}$, is a mapping from a delayed sequence of channel outputs $\mathbf{y}[i+\mathtt{T}]$, $\mathbf{y}[i+\mathtt{T}-1]$, …, and the corresponding channel states $\boldsymbol{\theta}[i+\mathtt{T}]$, $\boldsymbol{\theta}[i+\mathtt{T}-1]$, …, to an estimate of a source packet, $\mathbf{s}[i]$. For a particular channel state sequence, $\boldsymbol{\theta}$, an encoder, $\mathbf{C}$, and the associated decoder $\mathbf{C}^{-1}$, we define the decoding delay, $\mathbf{T}(\boldsymbol{\theta}, \mathbf{C}, n_c, \epsilon)$, as the minimum lag which guarantees successful decoding with probability of error at most $\epsilon$ when using packets of length $n_c$:

$$\mathbf{T}(\boldsymbol{\theta}, \mathbf{C}, n_c, \epsilon) \overset{\Delta}{=} \min_{\mathtt{T}} : \left\{\Pr\left[\mathbf{C}^{-1}\left(\mathbf{y}\left[{}^{i+\mathtt{T}}_{-\infty}\right], \boldsymbol{\theta}\left[{}^{i+\mathtt{T}}_{-\infty}\right]\right) \neq \mathbf{s}[i]\right] \leq \epsilon\right\} \tag{9.3}$$

where we use the notation $a\left[{}^{j}_{i}\right]$ to denote the subsequence $a[i]$, $a[i+1]$, …, $a[j]$.

## ■ 9.2.1 The Achievable Delay Region

In a broadcast or multicast scenario, different users receive the signal over different channel conditions. Similarly, even with only a single receiver, there may be some uncertainty about the channel. Traditionally in a static problem, this is modeled by defining a different channel probability law for each user or each possible channel. Instead, to obtain a dynamic model, we define a channel ensemble $\mathbf{\Theta}_i$ as a set of possible channel state sequences, $\boldsymbol{\theta}[i]$. The packet loss channel in the introduction provides one example of such a model. For another example, the channel ensemble $\mathbf{\Theta}_1$ could consist of all channel state sequences where five of the received packets have a signal-to-noise ratio up to 10 dB below nominal. Similarly, the channel ensemble $\mathbf{\Theta}_2$ could consist of all sequences where seven of the received packets have a signal-to-noise ratio up to 3 dB below nominal.

One natural performance measure for a code $\mathbf{C}$ on the channel ensemble $\mathbf{\Theta}$ is the maximum decoding delay for $\mathbf{C}$ on any state sequence in $\mathbf{\Theta}$:

$$\mathbf{T}_{\max}(\mathbf{\Theta}, \mathbf{C}, n_c, \epsilon) \overset{\Delta}{=} \max_{\boldsymbol{\theta} \in \mathbf{\Theta}} \mathbf{T}(\boldsymbol{\theta}, \mathbf{C}, n_c, \epsilon). \tag{9.4}$$

To model scenarios where the channel may behave in one of $N$ distinct modes (*e.g.*, $N$ users each receiving the transmission over different physical conditions), we can consider a collection of $N$ channel ensembles $\{\mathbf{\Theta}_1, \mathbf{\Theta}_2, \ldots, \mathbf{\Theta}_N\}$. In this case, performance can be measured by the delay tuple $(\mathtt{T}_1, \mathtt{T}_2, \ldots, \mathtt{T}_N)$ where $\mathtt{T}_i = \mathbf{T}_{\max}(\mathbf{\Theta}_i, \mathbf{C}, n_c, \epsilon)$.

We define the set of all possible delay tuples as the achievable delay region, as illustrated in Fig. 9-2 for $N = 2$. Determining the $N$-dimensional achievable delay region for arbitrary channel ensembles would provide the most complete understanding of delay in the packet-streaming model. Codes which achieve the best delay trade-offs characterized by such a region would be useful building blocks in designing practical systems. In this chapter, we focus on simpler performance measures corresponding to particular delay tuples.

To obtain a simple performance measure, we need to map the collection of channel ensembles $\{\mathbf{\Theta}_i\}_{i=1}^{N}$ to a single measure of overall delay efficiency. How should we construct this mapping? First, the measure should not completely ignore performance for any channel ensemble $\mathbf{\Theta}_i$. Second, the measure should lead to the design of codes which are good for a large collection of channel conditions. Finally, the measure should be reasonably simple to analyze.

Worst-case performance would seem to be a natural metric. Specifically, define the minimax-delay for a channel ensemble $\mathbf{\Theta}$ as the worst-case delay for the best code:

$$\mathbf{T}_{\max}^{\min}(\mathbf{\Theta}, n_c, \epsilon) \overset{\Delta}{=} \min_{\mathbf{C}} \max_{\boldsymbol{\theta} \in \mathbf{\Theta}} \mathbf{T}(\boldsymbol{\theta}, \mathbf{C}, n_c, \epsilon). \tag{9.5}$$

The codes labeled $\mathbf{C}_1$, $\mathbf{C}_2$, and $\mathbf{C}_{12}$ in Fig. 9-2 correspond to points achieving the minimax-delays $\mathbf{T}_{\max}^{\min}(\mathbf{\Theta}_1, n_c, \epsilon)$, $\mathbf{T}_{\max}^{\min}(\mathbf{\Theta}_2, n_c, \epsilon)$ and $\mathbf{T}_{\max}^{\min}(\mathbf{\Theta}_1 \cup \mathbf{\Theta}_2, n_c, \epsilon)$ respectively.

The drawback of using

$$\mathbf{T}_{\max}^{\min}(\mathbf{\Theta}_i, n_c, \epsilon)$$

as a performance measure is that it completely ignores performance on all channel ensembles except $\mathbf{\Theta}_i$. The drawback of using $\mathbf{T}_{\max}^{\min}\left(\cup_{i=1}^{N}\mathbf{\Theta}_i, n_c, \epsilon\right)$ is that this quantity is essentially determined by the worst channel in the collection $\cup_{i=1}^{N}\mathbf{\Theta}_i$. Hence codes designed to optimize $\mathbf{T}_{\max}^{\min}\left(\cup_{i=1}^{N}\mathbf{\Theta}_i, n_c, \epsilon\right)$ will not necessarily perform well on most of the ensembles in the collection.

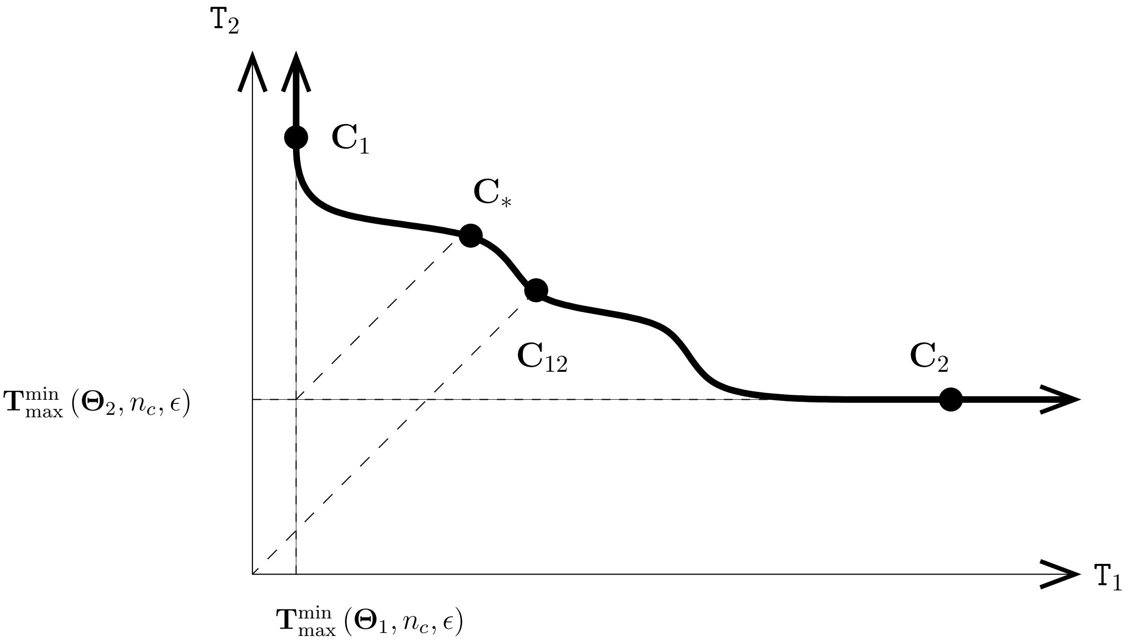

$T_2$
$\mathbf{C}_1$
$\mathbf{C}_*$
$\mathbf{C}_{12}$
$\mathbf{C}_2$
$\mathbf{T}_{\max}^{\min}(\mathbf{\Theta}_2, n_c, \epsilon)$
$T_1$
$\mathbf{T}_{\max}^{\min}(\mathbf{\Theta}_1, n_c, \epsilon)$

**Figure 9-2.** Conceptual illustration of a possible delay region. The axes represent delay for two channels ensembles $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$. Points above and to the right of the solid curve correspond to achievable delays. The points marked $\mathbf{C}_1$ and $\mathbf{C}_2$ correspond to codes achieving the minimum delay for $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$, while the code $\mathbf{C}_{12}$ minimizes the maximum delay. The code $\mathbf{C}_*$ minimizes the maximum difference between the delay for a fixed code and the best code for each channel ensemble.

To illustrate the drawback of worst-case delay for a concrete example, consider a congestion model where packet losses may occur. Let each $\mathbf{\Theta}_i$ represent the ensemble of packet loss patterns with exactly $i$ lost packets. The worst-case delay will always be determined by a packet loss pattern consisting of the maximum number of lost packets, *i.e.*,

$$\max_{\boldsymbol{\theta} \in \cup_{i=1}^{N} \mathbf{\Theta}_i} \mathbf{T}(\boldsymbol{\theta}, \mathbf{C}, n_c, \epsilon) = \max_{\boldsymbol{\theta} \in \mathbf{\Theta}_N} \mathbf{T}(\boldsymbol{\theta}, \mathbf{C}, n_c, \epsilon). \tag{9.6}$$

Hence $\mathbf{T}_{\max}^{\min}\left(\cup_{i=1}^{N} \mathbf{\Theta}_i, n_c, \epsilon\right)$ completely ignores performance for every channel ensemble except $\mathbf{\Theta}_N$ and only focuses on the worst channel conditions. Specifically, the worst-case delay does not consider performance when fewer losses occur: if only $N/2$ packet losses occur, then $\mathbf{C}$ could have a much lower decoding delay than $\mathbf{C}'$, but this is not captured by our performance metric. Ideally, we desire a code which can correct the loss of fewer packets with correspondingly shorter delay.

Intuitively, we would like a "universal" code which has the same delay as an optimal code designed for $\mathtt{B}$ packet losses when $\mathtt{B}$ packet losses occur and has the same delay as an optimal code designed for $\mathtt{B}'$ packet losses when $\mathtt{B}'$ occur. Thus, we believe that a valuable performance measure for a code is the codes deviation from being universal. We define the excess delay $\mathbf{T}_{\text{ex}}(\mathbf{C}, \{\mathbf{\Theta}_i\})$ of a code $\mathbf{C}$ as how much more delay $\mathbf{C}$ requires for each ensemble than a minimax delay optimal code designed specifically for that ensemble:

$$\mathbf{T}_{\text{ex}}(\mathbf{C}, \{\mathbf{\Theta}_i\}, n_c, \epsilon) \triangleq \max_i \left[\max_{\boldsymbol{\theta} \in \mathbf{\Theta}_i} \mathbf{T}(\boldsymbol{\theta}, \mathbf{C}, n_c, \epsilon) - \mathbf{T}_{\max}^{\min}(\mathbf{\Theta}_i, n_c, \epsilon)\right]. \tag{9.7}$$

The code $\mathbf{C}_*$ in Fig. 9-2 is an example of a code which minimizes the excess delay. While a code minimizing $\mathbf{T}_{\max}^{\min}\left(\cup_{i=1}^{N}\mathbf{\Theta}_i, n_c, \epsilon\right)$ is found by drawing a 45 degree line from the origin to the frontier of the achievable delay region, a code minimizing $\mathbf{T}_{\text{ex}}\left(\mathbf{C}, \{\mathbf{\Theta}_i\}, n_c, \epsilon\right)$ can be found by drawing a 45 degree line from the intersection of the $\mathbf{T}_{\max}^{\min}\left(\mathbf{\Theta}_i, n_c, \epsilon\right)$ hyper-planes to the frontier of the achievable region.

If the excess delay for a code $\mathbf{C}$ is zero, we call $\mathbf{C}$ delay-universal over the collection of channel ensembles $\{\mathbf{\Theta}_i\}$. Of course, whether such a universal code exists depends upon the collection of channel ensembles. In the sequel, we show that in many practical cases of interest such codes do exist and lead to systems which are good for a variety of channel conditions.

## ■ 9.3 Coding Theorems

In order to state our main result about the existence of delay-universal codes, we need the following definitions.

**Definition 3.** *A collection of channel ensembles* $\{\mathbf{\Theta}_i\}$ *is defined as* **input compatible** *if there exists an input probability distribution which simultaneously maximizes the mutual information* $I(x; y|\theta\,[i])$ *for every state* $\theta\,[i]$ *in every sequence* $\theta \in \mathbf{\Theta}_i$*.*

A collection of channel ensembles containing additive white Gaussian noise (AWGN) channels with different signal-to-noise ratios is an example of an input compatible collection while one containing both additive white Gaussian noise channels and additive exponential noise channels is not.[1] Intuitively, this property is important if universal codes are desired. Otherwise a code matched to one channel ensemble will have a sub-optimal mutual information on another channel ensemble and hence fail to be universally optimal. For non input compatible collections, techniques from the theory of compound channels and broadcast channels will be required.

**Definition 4.** *A channel ensemble* $\mathbf{\Theta}$ *is defined as* **permutation invariant** *if, for every state sequence* $\boldsymbol{\theta} \in \mathbf{\Theta}$*, every permutation of* $\boldsymbol{\theta}$ *is also in* $\mathbf{\Theta}$*.*

A channel ensemble where any `B` packets may be completely lost while the others are received error free and an ensemble where any `B` packets may be received with half the normal signal-to-noise ratio are both permutation invariant. Bursty channel ensembles like those in Chapter 7 where any consecutive `B` packets may be completely lost or received with half the normal signal-to-noise ratio are not permutation invariant.

Our main result, proved in Appendix G, is the following theorem:

**Theorem 28.** *If* $\{\mathbf{\Theta}_i\}$ *is a collection of input compatible, permutation invariant channel ensembles, then there exists a delay-universal code for* $\{\mathbf{\Theta}_i\}$*. Specifically, there exists a code,* $\mathbf{C}$*, such that for every* $\epsilon > 0$*, the excess delay is exactly zero for large enough packets:*

$$\forall n_c > n_0(\epsilon),\ \mathbf{T}_{\text{ex}}\left(\mathbf{C}, \{\mathbf{\Theta}_i\}, n_c, \epsilon\right) = 0. \tag{9.8}$$

[1] AWGN channels are also stochastically degraded [46]. In general, stochastically degraded channels are not necessarily input compatible, and input compatible channels are not necessarily stochastically degraded.

**A Packet Loss Example** To apply this result to streaming in a packet network, we can choose $\mathbf{\Theta}_i$ as the ensemble where at most $i$ packets are lost. For any $N$, $\{\mathbf{\Theta}_i\}_{i=1}^N$ is a collection of input compatible, permutation invariant channel ensembles and therefore Theorem 28 guarantees the existence of a universal code. In practice the minimum packet size, $n_0(\epsilon)$, need not be excessively large and codes requiring packet sizes proportional to $N/\log\epsilon$ with encoding and decoding times polynomial in the packet size can be easily constructed as shown in Section 9.4. Such lengths are well within the packet sizes used for Internet transmission as well as various wireless standards. For arbitrary channel models, tools from the traditional analysis of error exponents can be used to bound the required packet size.

The construction of the desired code is straightforward and uses standard random coding arguments. The key variation is that the code is designed to work in a packet-streaming model instead of a block model. The main effort required to prove the theorem is showing that no code designed for a specific $\mathbf{\Theta}_i$ can do better. To show this we need a tool to study the decoding delay of a code as a function of the channel mutual information.

## ■ 9.3.1 Information Debt

We define the mutual information debt at time $i$, $I_d[i, \mathtt{R}|\boldsymbol{\theta}]$, according to the recursion

$$I_d[i, \mathtt{R}|\boldsymbol{\theta}] = n_c \cdot \mathtt{R} - I(\mathbf{x}[i]\,;\mathbf{y}[i]\,|\theta[i]) + \max\{I_d[i-1, \mathtt{R}|\boldsymbol{\theta}], 0\} \tag{9.9}$$

with the added condition that $I_d[i_{\text{start}}, \mathtt{R}|\boldsymbol{\theta}] = 0$ where $i_{\text{start}}$ is the time of the initial transmission. The information debt can be thought of as how much more information we need to receive about the current messages before successful decoding can occur. This interpretation can be immediately translated into the following simple bound on decoding delay proved in Appendix G.

**Lemma 6.** *Consider a channel state sequence where the mutual information debt becomes positive at time $i_s$ and stays positive until at least time $i_f$. If $N$ is the largest integer below $I_d[i_f, \mathtt{R}|\boldsymbol{\theta}]/(n_c\mathtt{R})$, then at least one of the packets $\mathbf{s}[i_s]$, $\mathbf{s}[i_s+1]$, ..., $\mathbf{s}[i_f - N]$, transmitted by a causal, rate* R *system will fail to be decoded by time $i_f$.*

Note that Lemma 6 is a somewhat weak characterization of the decoding delay. Specifically, it tells us that as long as the information debt is positive at least *some* source packet can not be decoded, but it does not tell us exactly *which* packet can not be decoded. For example, imagine that a number of packets are lost starting at time $i_s$ and thus the information debt becomes positive at time $i_s$ and stays positive until time $i_f$. If the encoder somehow anticipated the packet losses it may be possible that at time $i_f$, the decoder can reconstruct all packets except the one at time $i_f - N$ and hence the decoding delay may be only $N$ instead of $i_f - i_s$.

Thus, in general, it is possible for the decoding delay required by a code to be less than the time the information debt is positive (*e.g.*, with burst erasure correction Chapter 7, [120, 119]). For a permutation invariant channel ensemble, however, this is not the case.

**Lemma 7.** *For any rate* R *code,* $\mathbf{C}$*, and permutation invariant channel ensemble,* $\mathbf{\Theta}$*, the worst-case decoding delay is at least the maximum possible time the information debt can be positive:*

$$\max_{\boldsymbol{\theta}\in\mathbf{\Theta}} \min_{\mathbf{C}} \mathtt{T}(\boldsymbol{\theta}, \mathbf{C}) \geq \max_{\boldsymbol{\theta}\in\mathbf{\Theta}} \sum_{i=-\infty}^{\infty} \Big[ I_d[i, \mathtt{R}|\boldsymbol{\theta}] \Big]_1 \tag{9.10}$$

*where* $\left[t\right]_1$ *is 1 for* $t > 0$ *and 0 otherwise.*

At this point we provide a sketch of the argument for Lemma 7 only for packet loss channels. The full proof for arbitrary channels is essentially the same but requires some tedious bookkeeping and is therefore relegated to Appendix G.

*Proof sketch for Lemma 7 on packet loss channels:* Let $\mathbf{\Theta}$ consist of all channel state sequences where exactly $\mathtt{B}$ packets may be completely lost and all others are correctly received. Assume for the sake of contradiction that Lemma 7 is false and the decoding delay for some code $\mathbf{C}$ is strictly less than the right hand side of (9.10) which we denote as $\mathtt{T}$. Consider the channel state sequence

$$\boldsymbol{\theta}^* = \theta^*\left[0\right], \theta^*\left[1\right], \ldots, \theta^*\left[\mathtt{T}-1\right] = 1^{\mathtt{B}}0^{\mathtt{T}-\mathtt{B}}$$

where $\theta^*\left[i\right] = 1$ indicates packet $i$ was erased.

By assumption, $\mathsf{s}\left[0\right]$ can be recovered by time $\mathtt{T}-1$ even though the information debt is positive at time $\mathtt{T}-1$. Now imagine that $\theta^*\left[\mathtt{T}\right]$ is erased. Since, $\mathsf{s}\left[0\right]$ is already recovered, the decoding delay for $\boldsymbol{\theta}^*$ is the same as for

$$\boldsymbol{\theta}^{**} = \theta^{**}\left[0\right], \theta^{**}\left[1\right], \ldots, \theta^{**}\left[\mathtt{T}\right] = 01^{\mathtt{B}-1}0^{\mathtt{T}-\mathtt{B}}1.$$

But since $\mathbf{\Theta}$ is permutation invariant $\boldsymbol{\theta}^{**} \in \mathbf{\Theta}$ and hence, by assumption, for $\boldsymbol{\theta}^{**}$, $\mathsf{s}\left[1\right]$ can be decoded at time $\mathtt{T}$. Since the decoding delay for $\boldsymbol{\theta}^*$ must be no worse than for, $\boldsymbol{\theta}^{**}$, $\mathsf{s}\left[1\right]$ can be decoded at time $\mathtt{T}$ for $\boldsymbol{\theta}^*$ even though

$$\mathsf{I}_d[\mathtt{T}, \mathtt{R}|\boldsymbol{\theta}^*] > \mathsf{I}_d[\mathtt{T}-1, \mathtt{R}|\boldsymbol{\theta}^*] > 0.$$

By continuing this argument, we can construct a channel state sequence with information debt greater than $\mathtt{T} \cdot n_c \cdot \mathtt{R}$ at time $i^*$, which, by assumption, can be decoded with delay $\mathtt{T}-1$. This contradicts Lemma 6 and shows our assumption that Lemma 7 was false is incorrect. □

## ■ 9.3.2 Random Code Constructions

Lemma 7 provides a lower bound on the decoding delay via the information debt. We present a random code construction which results in successful decoding whenever the mutual information debt is non-positive. Together these results can be used to characterize the connection between information debt and delay.

**Encoder Construction:** To construct an encoder we require a random partition, $\mathcal{H}(\cdot)$, mapping arbitrary length sequences from the alphabet $\mathcal{S}^{n_s}$ to random i.i.d. sequences of length $n_c$ from the alphabet $\mathcal{X}$. Specifically, $\mathcal{H}(\cdot)$ is generated according to the following procedure. For each element of the alphabet $\mathcal{S}^{n_s}$, independently select $n_c$ elements from the alphabet $\mathcal{X}$ using the distribution $p(x)$. Repeat this procedure for each element of the alphabets $\mathcal{S}^{2n_s}$, $\mathcal{S}^{3n_s}$, etc.

This code structures differs from the one in Chapter 7 in that the former essentially provides an equal amount of redundancy from past packets into packet $i$ to provide robustness to a wide range of potential channel dynamics. In contrast, the code structure in Chapter 7 provides more redundancy from packets further in the past to tailor the code to the specific burst dynamics.

**Encoding and Decoding:** The initial source packet, $\mathbf{s}[0]$, is encoded to form $\mathbf{x}[0] = \mathcal{H}(\mathbf{s}[0])$. The next source packet, $\mathbf{s}[1]$, is encoded to form $\mathbf{x}[1] = \mathcal{H}(\mathbf{s}[0], \mathbf{s}[1])$, the one after that is encoded to form $\mathbf{x}[2] = \mathcal{H}(\mathbf{s}[0], \mathbf{s}[1], \mathbf{s}[2])$, etc. To decode $\mathbf{s}[i]$ for any $\epsilon > 0$, the decoder waits until $I_d[i+j, \mathtt{R}+\epsilon|\boldsymbol{\theta}] \leq 0$. Then the decoder searches for a unique sequence of packets $\ldots$, $\mathbf{s}[i]$, $\mathbf{s}[i+1]$, $\ldots$, $\mathbf{s}[i+j]$ such that $\ldots$, $\mathbf{x}[i]$, $\mathbf{x}[i+1]$, $\ldots$, $\mathbf{x}[i+j]$ are jointly typical with $\ldots$, $\mathbf{y}[i]$, $\mathbf{y}[i+1]$, $\ldots$, $\mathbf{y}[i+j]$. If no such unique sequence is found, the decoder declares a decoding failure.

**Probability of Error:**

**Lemma 8.** *For any $\epsilon > 0$, there exists an $n_0$ such that for all $n_c > n_0$, the probability of decoding failure is less than $\epsilon$.*

*Proof.* We prove the Lemma using standard arguments. Let $\mathcal{E}$ be the event that the decoder fails. The event $\mathcal{E}$ can be separated into the event that the transmitted codeword is not jointly typical with the received sequence (denoted by $\mathcal{E}_1$) and the event that an incorrect codeword is jointly typical with the received sequence (denoted by $\mathcal{E}_2$). The union bound implies $\Pr[\mathcal{E}] \leq \Pr[\mathcal{E}_1] + \Pr[\mathcal{E}_2]$. By the law of large numbers, $\Pr[\mathcal{E}_1] \to 0$ as $n_c \to \infty$, so all that remains is to bound $\Pr[\mathcal{E}_2]$.

Note that if $I_d[i+j, \mathtt{R}+\epsilon|\boldsymbol{\theta}] \leq 0$, then according to (9.9),

$$I\left(\mathbf{x}\left[{}^{i+j}_{0}\right]; \mathbf{y}\left[{}^{i+j}_{0}\right] | \boldsymbol{\theta}\left[{}^{i+j}_{0}\right]\right) \geq (i+j-1) \cdot n_c(\mathtt{R}+\epsilon). \tag{9.11}$$

This leads to the following chain of inequalities

$$\begin{aligned}
\Pr[\mathcal{E}] &\leq |\mathcal{S}|^{n_s \cdot (i+j-1)} \cdot \exp\left\{-I\left(\mathbf{x}\left[{}^{i+j}_{0}\right]; \mathbf{y}\left[{}^{i+j}_{0}\right] | \boldsymbol{\theta}\left[{}^{i+j}_{0}\right]\right)\right\} &(9.12)\\
&= \exp\left\{(i+j-1) \cdot n_s \log|\mathcal{S}| - I\left(\mathbf{x}\left[{}^{i+j}_{0}\right]; \mathbf{y}\left[{}^{i+j}_{0}\right] | \boldsymbol{\theta}\left[{}^{i+j}_{0}\right]\right)\right\} &(9.13)\\
&\leq \exp\left\{(i+j-1) \cdot \frac{n_s}{n_c} \cdot n_c \log|\mathcal{S}| - n_c \cdot (i+j-1)(\mathtt{R}+\epsilon)\right\} &(9.14)\\
&= \exp\left\{-n_c \cdot (i+j-1) \cdot \epsilon\right\}. &(9.15)
\end{aligned}$$

The right hand side of (9.12) consists of the number of possible codewords times a bound on the probability that any incorrect codeword is typical with the received sequence. We obtain (9.14) from (9.11), and (9.15) comes from our definition of rate (9.2). □

We can now use these Lemmas to prove Theorem 28.

*Proof of Theorem 28:* Since the collection of channel ensembles is input compatible, the preceding code construction yields a code where the decoding delay is given by the length of time the information debt stays positive. According to Lemma 7, for any particular channel ensemble, $\boldsymbol{\Theta}_i$, the minimax decoding delay for any code is given by the maximum time the information debt can stay positive. Hence the preceding code construction is universal. □

## ■ 9.4 Code Constructions

In this section we consider convolutional code constructions designed to recover from erasures as quickly as possible. In contrast to previous constructions focusing on maximizing the total number of correctable erasures regardless of delay [166], our chief interest is not optimizing the free distance. Instead, we sketch a construction designed to obtain codes which can successfully decode whenever the mutual information debt returns to 0.

While Section 9.3 proves the existence of infinite memory encoders with fast decoding properties, an important constraint for practical systems is a finite encoder memory. For a memory $\mathtt{M}$ system where the encoder output $\boldsymbol{\mathsf{x}}[i]$ must be a function of $\boldsymbol{\mathsf{s}}[i]$, $\boldsymbol{\mathsf{s}}[i-1]$, …, $\boldsymbol{\mathsf{s}}[i-\mathtt{M}]$, successful decoding will only be possible if $I_d[i, \mathtt{R}|\boldsymbol{\theta}]$ stays positive for at most $\mathtt{M}$ packets. We sketch a linear, periodically time-varying (LPTV) code construction which has this property.

Linear encoders are traditionally specified via a generator matrix, $\mathbf{G}(D)$, where $D$ represents the delay operator [87]. However, we find it more convenient to consider the encoder as an infinite matrix operating on the input via

$$\boldsymbol{\mathsf{x}} = \mathbf{C}\boldsymbol{\mathsf{s}}.$$

Our notation and encoding system is best illustrated by example. Hence, we focus on describing a memory $\mathtt{M} = 2$, rate $k/n = 2/3$ encoder where the output is obtained by multiplying the input by a block Vandermonde matrix

$$\begin{pmatrix} \mathsf{x}_0[0] \\ \mathsf{x}_1[0] \\ \mathsf{x}_2[0] \\ \mathsf{x}_0[1] \\ \mathsf{x}_1[1] \\ \mathsf{x}_2[1] \\ \mathsf{x}_0[2] \\ \mathsf{x}_1[2] \\ \mathsf{x}_2[2] \\ \mathsf{x}_0[3] \\ \mathsf{x}_1[3] \\ \mathsf{x}_2[3] \\ \mathsf{x}_0[4] \\ \mathsf{x}_1[4] \\ \mathsf{x}_2[4] \\ \vdots \end{pmatrix} = \begin{pmatrix} z_1^4 & z_1^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ z_2^4 & z_2^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ z_3^4 & z_3^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ z_4^2 & z_4^3 & z_4^4 & z_4^5 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ z_5^2 & z_5^3 & z_5^4 & z_5^5 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ z_6^2 & z_6^3 & z_6^4 & z_6^5 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 1 & z_7^1 & z_7^2 & z_7^3 & z_7^4 & z_7^5 & 0 & 0 & 0 & 0 & \cdots \\ 1 & z_8^1 & z_8^2 & z_8^3 & z_8^4 & z_8^5 & 0 & 0 & 0 & 0 & \cdots \\ 1 & z_9^1 & z_9^2 & z_9^3 & z_9^4 & z_9^5 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & z_1^1 & z_1^2 & z_1^3 & z_1^4 & z_1^5 & 0 & 0 & \cdots \\ 0 & 0 & 1 & z_2^1 & z_2^2 & z_2^3 & z_2^4 & z_2^5 & 0 & 0 & \cdots \\ 0 & 0 & 1 & z_3^1 & z_3^2 & z_3^3 & z_3^4 & z_3^5 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 1 & z_4^1 & z_4^2 & z_4^3 & z_4^4 & z_4^5 & \cdots \\ 0 & 0 & 0 & 0 & 1 & z_5^1 & z_5^2 & z_5^3 & z_5^4 & z_5^5 & \cdots \\ 0 & 0 & 0 & 0 & 1 & z_6^1 & z_6^2 & z_6^3 & z_6^4 & z_6^5 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \ddots \end{pmatrix} \cdot \begin{pmatrix} \mathsf{s}_0[0] \\ \mathsf{s}_1[0] \\ \mathsf{s}_0[1] \\ \mathsf{s}_1[1] \\ \mathsf{s}_0[2] \\ \mathsf{s}_1[2] \\ \mathsf{s}_0[3] \\ \mathsf{s}_1[3] \\ \mathsf{s}_0[4] \\ \mathsf{s}_1[4] \\ \vdots \end{pmatrix} \qquad (9.16)$$

where the $z_i$'s represent elements in a finite field which are yet to be specified. Note that the memory and causality of the encoder is evident from the block structure of $\mathbf{C}$. The zeros in the upper right are due to the causality restriction while the zeros in the lower left are due to the requirement that the encoder output for packet $i$ can only depend on packets $i$, $i-1$ and $i-2$.

For a rate $k/n$ code with memory $\mathtt{M}$, an LPTV code with period $\mathtt{M}+1$ could be constructed along similar lines by creating $\mathtt{M}+1$ Vandermonde matrices each of dimension $n$ by $k\cdot(\mathtt{M}+1)$. The infinite matrix $\mathbf{C}$ would be created by using shifted copies of these Vandermonde matrices repeated periodically. Comparing the code structure above to the burst correcting code constructions in Chapter 8 shows that the former essentially treats past packets equally while the latter treats past packets differently as required to match the bursty channel dynamics.

Returning to the rate 2/3 example, suppose that all symbols are received correctly except that $\boldsymbol{\mathsf{x}}[1]$ is erased. The information debt becomes positive at time $i = 1$ and returns to zero at time $i = 3$. To decode at time $i = 3$ we consider the relationship between $\boldsymbol{\mathsf{x}}[i]$

and $\mathbf{s}[i]$ given by the matrix equation

$$\begin{pmatrix} \mathsf{x}_0[2] \\ \mathsf{x}_1[2] \\ \mathsf{x}_2[2] \\ \mathsf{x}_0[3] \\ \mathsf{x}_1[3] \\ \mathsf{x}_2[3] \end{pmatrix} = \begin{pmatrix} z_7^2 & z_7^3 & z_7^4 & z_7^5 & 0 & 0 \\ z_8^2 & z_8^3 & z_8^4 & z_8^5 & 0 & 0 \\ z_9^2 & z_9^3 & z_9^4 & z_9^5 & 0 & 0 \\ 1 & z_1^1 & z_1^2 & z_1^3 & z_1^4 & z_1^5 \\ 1 & z_2^1 & z_2^2 & z_2^3 & z_2^4 & z_2^5 \\ 1 & z_3^1 & z_3^2 & z_3^3 & z_3^4 & z_3^5 \end{pmatrix} \cdot \begin{pmatrix} \mathsf{s}_0[1] \\ \mathsf{s}_1[1] \\ \mathsf{s}_0[2] \\ \mathsf{s}_1[2] \\ \mathsf{s}_1[3] \\ \mathsf{s}_2[3] \end{pmatrix}. \tag{9.17}$$

Technically, the matrix in (9.17) should be a 9-by-6 matrix which operates on a column vector containing $\mathbf{s}[0]$ through $\mathbf{s}[3]$, but since $\mathbf{s}[0]$ was received correctly by assumption, we can always subtract out the effect of terms before the erasure.

Successful decoding will be possible if we can invert the equation in (9.17) to recover $\mathbf{s}[i]$ from $\mathbf{x}[i]$ or equivalently if the matrix in (9.17) is non-singular. If the zeros were not present and we had a true Vandermonde matrix, then a Vandermonde argument [42] could be used to prove the determinant is non-zero when $z_i \neq z_j$ for $i \neq j$. Instead, to check if this matrix is singular we can perform Gaussian elimination to zero out terms above the diagonal using the algorithm in Table 9.1.

**Table 9.1.** The algorithm **VANDERMONDE-ELIMINATE(C)**.

```
$(N, M) \leftarrow \text{size}(\mathbf{C})$
$r \leftarrow 1$
while $(r \leq N)$ do
   $c \leftarrow M$
   while $(c > r)$ do
      if $(\mathbf{C}(r, c-1) \neq 0)$ then
         $\mathbf{C}(1:N, c) \leftarrow \mathbf{C}(1:N, c) - \mathbf{C}(1:N, c-1) \cdot \frac{\mathbf{C}(r,c)}{\mathbf{C}(r,c-1)}$
      end if
      $c \leftarrow c - 1$
   end while
   $r \leftarrow r + 1$
end while
```

The result of running this algorithm on the matrix in (9.17) for one iteration of the outer loop produces the following result

$$\begin{pmatrix} z_7^2 & 0 & 0 & 0 & 0 & 0 \\ z_8^2 & z_8^3 + p_2(z_7, z_8) & z_8^4 + p_3(z_7, z_8) & z_8^5 + p_4(z_7, z_8) & 0 & 0 \\ z_9^2 & z_9^3 + p_2(z_7, z_9) & z_9^4 + p_3(z_7, z_9) & z_9^5 + p_4(z_7, z_9) & 0 & 0 \\ 1 & z_1 + p_0(z_7, z_1) & z_1^2 + p_1(z_7, z_1) & z_1^3 + p_2(z_7, z_1) & z_1^4 + p_3(z_7, z_1) & z_1^5 + p_4(z_7, z_1) \\ 1 & z_2 + p_0(z_7, z_2) & z_2^2 + p_1(z_7, z_2) & z_2^3 + p_2(z_7, z_2) & z_2^4 + p_3(z_7, z_2) & z_2^5 + p_4(z_7, z_2) \\ 1 & z_3 + p_0(z_7, z_3) & z_3^2 + p_1(z_7, z_3) & z_3^3 + p_2(z_7, z_3) & z_3^4 + p_3(z_7, z_3) & z_3^5 + p_4(z_7, z_3) \end{pmatrix}$$

while another iteration of the outer loop yields

$$\begin{pmatrix} z_7^2 & 0 & 0 & 0 & 0 & 0 \\ z_8^2 & z_8^3 & 0 & 0 & 0 & 0 \\ z_9^2 & z_9^3 + p_2(z_7, z_8, z_9) & z_9^4 + p_3(z_7, z_8, z_9) & z_9^5 + p_4(z_7, z_8, z_9) & 0 & 0 \\ 1 & z_1 + p_0(z_7, z_8, z_1) & z_1^2 + p_1(z_7, z_8, z_1) & z_1^3 + p_2(z_7, z_8, z_1) & z_1^4 + p_3(z_7, z_8, z_1) & z_1^5 + p_4(z_7, z_8, z_1) \\ 1 & z_2 + p_0(z_7, z_8, z_2) & z_2^2 + p_1(z_7, z_8, z_2) & z_2^3 + p_2(z_7, z_8, z_2) & z_2^4 + p_3(z_7, z_8, z_2) & z_2^5 + p_4(z_7, z_8, z_2) \\ 1 & z_3 + p_0(z_7, z_8, z_3) & z_3^2 + p_1(z_7, z_8, z_3) & z_3^3 + p_2(z_7, z_8, z_3) & z_3^4 + p_3(z_7, z_8, z_3) & z_3^5 + p_4(z_7, z_8, z_3) \end{pmatrix}$$

where the notation $p_i(\alpha, \beta, \ldots, \gamma)$ stands for some rational polynomial in $\alpha$, $\beta$, …, $\gamma$ of degree at most $i$ in the variable $\gamma$. These intermediate results illustrate two important invariants of the algorithm.

**Theorem 29.** *When the algorithm* **VANDERMONDE-ELIMINATE** *is called on a block Vandermonde matrix* $\mathbf{C}$*, the following conditions are true for every step of the algorithm:*

1. *For all $r \geq c$, the entry, $\mathbf{C}_{(r,c)}$, in row $r$ and column $c$, is a rational polynomial with a degree greater than rational polynomial in $\mathbf{C}_{(r,c-1)}$ with respect to the row variable $\alpha_r$. Furthermore the highest degree term of this polynomial has the coefficient 1.*

2. *Row $r$ does not contain any of the row variables in rows greater than $r$.*

*Proof.* The theorem can be proved by induction. A block Vandermonde matrix satisfies the required invariants by construction (this is the base case). Next we assume that the invariants hold at some iteration of the algorithm and prove that they hold at the next step. Since $\mathbf{C}$ is only modified inside the if statement, we only need to consider the subtracted term

$$\mathbf{C}(1 : N, c-1) \cdot \frac{\mathbf{C}(r, c)}{\mathbf{C}(r, c-1)}. \tag{9.18}$$

The term $\mathbf{C}(r, c)/\mathbf{C}(r, c-1)$ does not contain any of the row variables in rows greater than $r$ since condition 2 holds by assumption. Since $\mathbf{C}(1 : N, c-1)$ satisfies both invariants by assumption, the product in (9.18) satisfies the invariants for rows $r+1$ through $N$ and therefore so does $\mathbf{C}(j, c)$, for $r + 1 \leq j \leq N$ after the subtraction. Furthermore, $\mathbf{C}(j, c)$ must be 0 for $j \leq r$ after the subtraction since the algorithm performs Gaussian elimination. Therefore the second condition is satisfied for $j \leq r$ and hence all $j$. The first condition is trivially satisfied for $j \leq r$ since the test in the inner while loop implies $r < c$ and the first condition applies only to $\mathbf{C}(j, i)$ with $j \geq i$. □

According to this theorem, the final result of calling **VANDERMONDE-ELIMINATE** on the matrix in (9.17) will have the diagonal corresponding to

$$\begin{pmatrix} z_7^2 + p_1(z_7) \\ z_8^3 + p_3(z_7, z_8) \\ z_9^4 + p_3(z_7, z_8, z_9) \\ z_1^3 + p_2(z_7, z_8, z_9, z_1) \\ z_2^4 + p_3(z_7, z_8, z_9, z_1, z_2) \\ z_3^5 + p_4(z_7, z_8, z_9, z_1, z_2, z_3) \end{pmatrix} \tag{9.19}$$

and decoding will be successful provided that values for $z_i$ are chosen such that the product of terms in (9.19) is non-zero. While there may exist an efficient algorithm for choosing the values of $z_i$, we instead show that for large enough field sizes a random choice will result in successful decoding with high probability.

**Theorem 30.** *If the row variables, $z_i$, for a rate $k/n$ code with memory $\mathtt{M}$ are chosen uniformly at random over the non-zero elements of a finite field* $\mathrm{GF}(q)$*, then successful decoding will be possible if the information debt becomes positive and returns to zero within* $\mathtt{M}+1$ *packets except with probability at most*

$$\frac{n \cdot (\mathtt{M}+1)^2 \cdot 2^{n\cdot(\mathtt{M}+1)}}{q-1-n\cdot(\mathtt{M}+1)}. \tag{9.20}$$

*Proof.* Consider decoding of a particular erasure pattern. Successful decoding will be possible if the matrix, $\mathbf{C}$, corresponding to the encoding of the unerased symbols is non-singular. By the assumption that the mutual information debt returned to zero within $(\mathtt{M}+1)$ packets, there can be no more than $n \cdot (\mathtt{M}+1)$ rows and columns in this matrix. Let $\mathbf{V}$ be the result of calling **VANDERMONDE-ELIMINATE** on $\mathbf{C}$. Successful decoding will occur provided that none of the $n \cdot (\mathtt{M}+1)$ diagonal terms of $\mathbf{V}$ is 0. If we let $\mathbf{z}_{(\bar{j})}$ denote the vector of row variables with $z_j$ omitted then we have

$$\begin{aligned}
\Pr[\det(\mathbf{C})=0] &= \Pr\left[\cup_{j=1}^{n\cdot(\mathtt{M}+1)} \mathbf{V}_{(j,j)}=0\right] && (9.21)\\
&\le \sum_{j=1}^{n\cdot(\mathtt{M}+1)} \Pr[\mathbf{V}_{(j,j)}=0] && (9.22)\\
&= \sum_{j=1}^{n\cdot(\mathtt{M}+1)} \sum_{\mathbf{z}_{(\bar{j})}} \Pr[\mathbf{V}_{(j,j)}=0|\mathbf{z}_{(\bar{j})}=\mathbf{z}_{(\bar{j})}] \cdot \Pr[\mathbf{z}_{(\bar{j})}=\mathbf{z}_{(\bar{j})}] && (9.23)\\
&\le \sum_{j=1}^{n\cdot(\mathtt{M}+1)} \sum_{\mathbf{z}_{(\bar{j})}} \frac{1}{q-1-n\cdot(\mathtt{M}+1)} \cdot \Pr[\mathbf{z}_{(\bar{j})}=\mathbf{z}_{(\bar{j})}] && (9.24)\\
&= \frac{n\cdot(\mathtt{M}+1)}{q-1-n\cdot(\mathtt{M}+1)} && (9.25)
\end{aligned}$$

where (9.24) follows from Theorem 29 and the assumption that $z_j \neq 0$.

Since there are at most $2^{n\cdot(\mathtt{M}+1)}$ correctable erasure patterns starting at time $i$ and ending by time $i+\mathtt{M}$ and the code is periodic with period $\mathtt{M}+1$, there are at most

$$(\mathtt{M}+1)\cdot 2^{n\cdot(\mathtt{M}+1)} \tag{9.26}$$

distinct possibilities for $\mathbf{C}$. Thus the probability that a random choice of $z_i$'s yields a 0 determinant for any $\mathbf{C}$ is at most the product of (9.26) and (9.25) which yields (9.20). □

Thus to make the probability of choosing a bad code less than *e.g.*, $10^{-6}$, for our rate 2/3, memory 2 example we can apply Theorem 30 and select

$$q \ge \frac{n\cdot(\mathtt{M}+1)^2\cdot 2^{n\cdot(\mathtt{M}+1)}}{10^{-6}} + 1 + n\cdot(\mathtt{M}+1) \approx 1.382\cdot 10^{10}. \tag{9.27}$$

For example, choosing $q = 2^{34}$ (*i.e.*, choosing a finite field with 34-bit symbols) would suffice.

We emphasize that the bound on choosing a bad code developed in Section 30 is quite conservative. We believe that much tighter bounds can be developed for the random selection method and that simple modifications of the random selection method can significantly decrease the probability of choosing the $z_i$'s poorly.

## 9.5 Delay and Stability Analysis

Since the information debt characterizes when successful decoding can occur for causal systems, we can study the delay required as a fundamental property of the channel independent of a particular code structure. Specifically, if $\mathsf{I}_d[i, \mathtt{R}|\boldsymbol{\theta}]$ increases above 0 at time $i = i_s$, the decoder will be unable to completely decode the transmitted message until time $i_s + \mathsf{t}$ when $\mathsf{I}_d[i+\mathsf{t}, \mathtt{R}|\boldsymbol{\theta}]$ becomes non-positive. Furthermore, according to Theorem 28, there exist codes with delay corresponding to $\mathsf{t}$. Thus, a natural quantity of interest is the decoding delay $\mathsf{t}$.

Specifically, when delay is an issue, the ergodic Shannon capacity of a channel may not be relevant since achieving such a rate may require very long delays. Instead, one may define a more useful performance measure as the maximum rate at which data can be sent such that the probability of the decoding delay for a packet exceeding $\mathtt{T}$ is at most $\epsilon$. Such a definition is similar to the well-known outage-capacity formulation [141, 23]. Alternatively, since packets will generally have variable delays, one could consider the maximum rate at which the expected delay for a packet is at most $\mathtt{T}$. Computing such rates is analogous to analyzing stability and delay in queuing systems.

Intuitively, an important concern is whether the information transmitted will ever be decoded (or equivalently, whether the random variable $\mathsf{t}$ will always be finite). This corresponds to the analogous concern of whether incoming requests to a queue will ever be serviced or whether the waiting time random variable will always be finite. We know that if the transmitter sends information at a rate larger than the ergodic channel capacity then there is no hope of successful decoding. Thus $\mathsf{I}_d[i, \mathtt{R}|\boldsymbol{\theta}]$ will tend to drift upward such that there is no finite $\mathsf{t}$ where $\mathsf{I}_d[i + t, \mathtt{R}|\boldsymbol{\theta}]$ becomes non-positive. However, if the transmitter sends at a rate below the ergodic capacity it is slightly less obvious whether successful decoding will eventually occur or whether there is a chance that the delay will continue to grow without bound. Roughly speaking we refer to the case where $\mathsf{I}_d[i, r|\boldsymbol{\theta}]$ tends to grow to $\infty$ as an unstable system and the case where $\mathsf{I}_d[i, r|\boldsymbol{\theta}]$ tends to eventually become non-positive as a stable system.

We can analyze $\mathsf{t}$ and hence the stability of a causal system by considering the information debt as a random walk starting at time $i_s + 1$ and continuing until $\mathsf{I}_d[i, \mathtt{R}|\boldsymbol{\theta}]$ becomes non-positive. Specifically, we define

$$\mathsf{v}\,[i] \stackrel{\Delta}{=} \mathsf{I}_d[i + i_s, \mathtt{R}|\boldsymbol{\theta}] - \mathsf{I}_d[i + i_s - 1, \mathtt{R}|\boldsymbol{\theta}] \tag{9.28}$$

as the increment of the random walk defined by

$$\mathsf{w}[i] \stackrel{\Delta}{=} \sum_{j=1}^{i} \mathsf{v}\,[j]\,. \tag{9.29}$$

Note that the increment $\mathsf{v}\,[i]$ is the expected difference between the channel capacity and the transmission rate. The random variable $\mathsf{t}$ is defined as the earliest time when $\mathsf{w}[i]$ crosses a threshold at $-\mathsf{I}_d[i_s, \mathtt{R}|\boldsymbol{\theta}]$.

When the random process $\boldsymbol{\theta}\,[i]$ is i.i.d. and a semi-invariant moment generating function, $\gamma(r) = \log\{E[e^{r\cdot\mathsf{v}[i]}]\}$, exists in some open interval near the origin, we can use Wald's Identity [70, Chapter 7],

$$E[\exp(r \cdot \mathsf{w}[\mathsf{t}] - \mathsf{t} \cdot \gamma(r))] = 1, \tag{9.30}$$

to compute various properties of $\mathsf{t}$.

### Delay Bounds For i.i.d. Channels

**Theorem 31.** *Consider a channel where $\boldsymbol{\theta}[i]$ is an i.i.d. process such that the semi-invariant moment generating function $\gamma(r) = \log\{E[\exp r \cdot \mathsf{v}]\}$ exists for some open interval near the origin and $\mathtt{R} < E[\mathsf{I}(\mathsf{x};\mathsf{y})]$. If $\mathsf{I}_d[i, R|\boldsymbol{\theta}]$ becomes positive at $i = i_s$ then $\mathsf{I}_d[i, R|]$ first becomes non-positive at time $i_s + \mathsf{t}$ where*

$$\frac{R - E[\mathsf{I}(\mathsf{x};\mathsf{y})|\mathsf{I}(\mathsf{x};\mathsf{y}) < R]}{E[\mathsf{I}(\mathsf{x};\mathsf{y}) - R]} \leq E[\mathsf{t}] \leq \frac{R - E[\mathsf{I}(\mathsf{x};\mathsf{y})|\mathsf{I}(\mathsf{x};\mathsf{y}) < R]}{E[\mathsf{I}(\mathsf{x};\mathsf{y}) - R]} + \frac{\max[\mathsf{I}(\mathsf{x};\mathsf{y}) - R]}{E[\mathsf{I}(\mathsf{x};\mathsf{y}) - R]} \quad (9.31)$$

**Theorem 32.** *For the conditions of Theorem 31,*

$$\frac{\mathrm{VAR}[\mathsf{t}]}{E[\mathsf{t}]} \leq \frac{\mathrm{VAR}[\mathsf{I}(\mathsf{x};\mathsf{y}) - \mathtt{R}]}{E[\mathsf{I}(\mathsf{x};\mathsf{y}) - \mathtt{R}]^2} + \frac{\max \mathsf{I}(\mathsf{x};\mathsf{y}) - \mathtt{R}}{E[\mathsf{I}(\mathsf{x};\mathsf{y}) - \mathtt{R}]^2}. \quad (9.32)$$

In addition to the mean and variance of $\mathsf{t}$ we may be interested in the tails of the distribution for $\mathsf{t}$. The following theorem provides an exponential bound to $\mathsf{t}$.

**Theorem 33.** *For the conditions of Theorem 31 with $r^*$ being the smallest positive root of $\gamma(r)$,*

$$\Pr[\mathsf{t} > \mathtt{T}] \leq \exp[r \cdot (\mathsf{I}_d[i_s, \mathtt{R}|\boldsymbol{\theta}] + \max \mathsf{I}(\mathsf{x};\mathsf{y}) - \mathtt{R}) + \mathtt{T} \cdot \gamma(r)], \quad r < r^*. \quad (9.33)$$

*and*

$$\Pr[\mathsf{t} < \mathtt{T}] \leq \exp[r \cdot (\mathsf{I}_d[i_s, \mathtt{R}|\boldsymbol{\theta}] + \max \mathsf{I}(\mathsf{x};\mathsf{y}) - \mathtt{R}) + \mathtt{T} \cdot \gamma(r)], \quad r > r^*. \quad (9.34)$$

## ■ 9.5.1 An Erasure Channel Example

Next we consider a packet erasure channel example. Specifically, each transmitted packet $\mathbf{x}[i]$ is correctly received with probability $1 - p$ (in which case $\mathbf{y}[i] = \mathbf{x}[i]$) or erased with probability $p$ (in which case $\mathbf{y}[i] = \varnothing$). The capacity of such a channel is $1 - p$.

When erasures are i.i.d., the required quantities in Theorem 31 and Section 32 can be computed by noting that

$$\Pr[\mathsf{I}(\mathsf{x}[i];\mathsf{y}[i]) = 0] = p$$
$$\Pr[\mathsf{I}(\mathsf{x}[i];\mathsf{y}[i]) = 1] = 1 - p.$$

Therefore we have

$$\frac{\mathtt{R}}{1 - p - \mathtt{R}} \leq E[\mathsf{t}] \leq \frac{1}{1 - p - \mathtt{R}}$$

and

$$\frac{\mathrm{VAR}[\mathsf{t}]}{E[\mathsf{t}]} \leq \frac{2 - p - \mathtt{R}}{(1 - p) \cdot (1 - \mathtt{R})^2 + p \cdot \mathtt{R}^2}.$$

These quantities are plotted in Fig. 9-3. The figure illustrates that although the system has a finite expected delay (and is therefore stable) for every rate below the channel capacity, as the transmission rate approaches the channel capacity, the expected delay increases to infinity inversely with $C - \mathtt{R}$. Thus Fig. 9-3 illustrates that if we require the expected decoding delay to be bounded, the rate we can achieve is a fraction of the Shannon capacity.

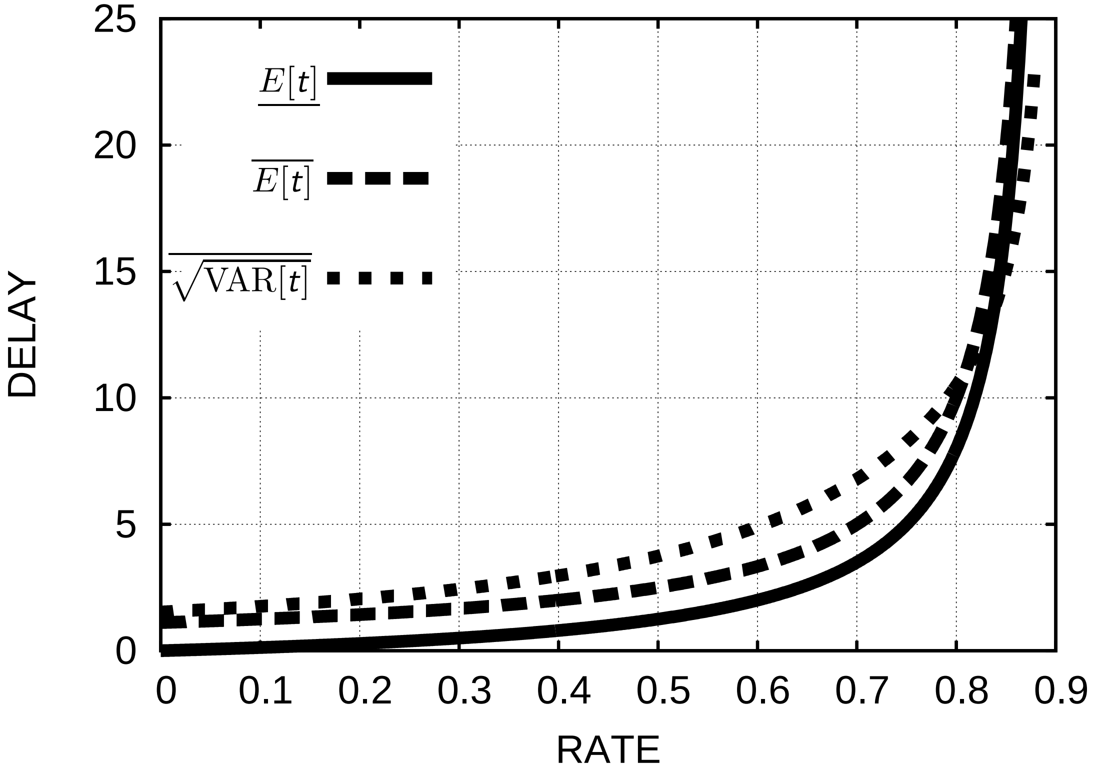


**Figure 9-3.** Bounds on decoding delay for an erasure channel. The plot illustrates upper bounds on the mean and standard deviation ($\overline{E[t]}$ and $\overline{\sqrt{\mathrm{VAR}[t]}}$) and a lower bound on the mean ($\underline{E[t]}$5) for the decoding delay for a channel with erasure probability of 10%.

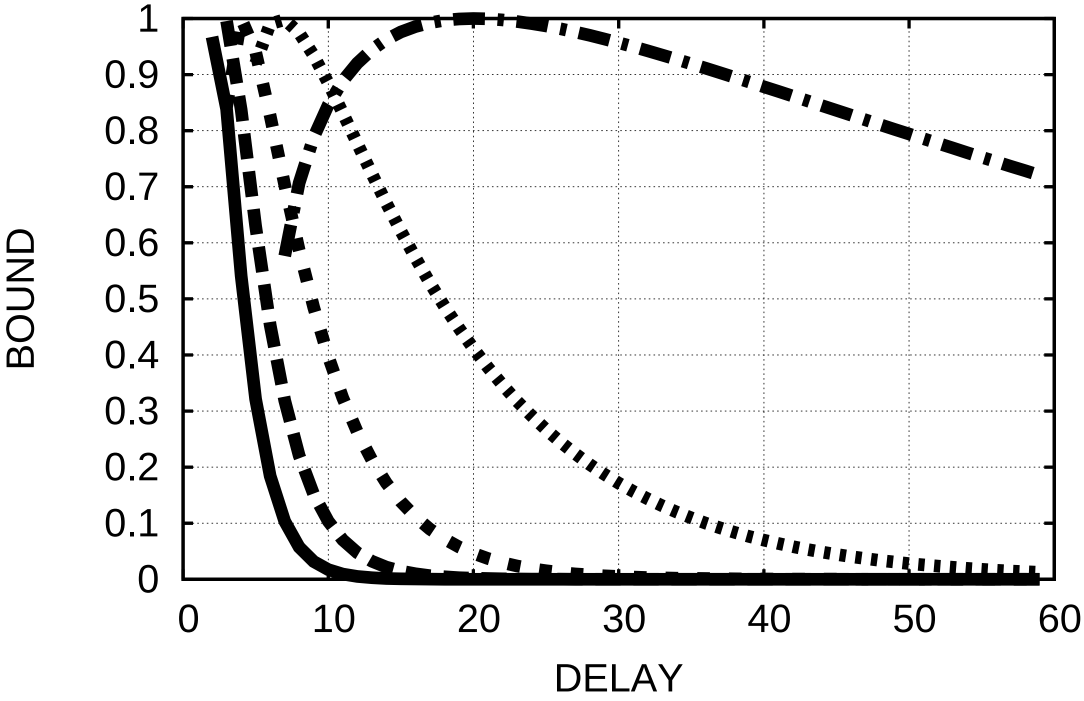


**Figure 9-4.** Bounds on the probability that the delay exceeds a given value for a channel with erasure probability $p = 0.1$. From left to right the plots correspond to $\mathtt{R} = .45$, .55, .65, .75, and .85.

To compute bounds on $\Pr[t > \mathtt{T}]$ from Theorem 33 we note that

$$\gamma(r) = \log E[\exp(r \cdot v)] = \log[(1-p) \cdot e^{r \cdot (R-1)} + pe^{r \cdot R}] \tag{9.35}$$

and therefore the right hand side of (9.33) and (9.34) is

$$\exp\left\{r + \mathtt{T} \cdot \log[(1-p) \cdot \exp(r \cdot (\mathtt{R}-1)) + p \cdot \exp(r \cdot \mathtt{R})]\right\}.$$

To optimize the parameter $r$ in the bound we take the logarithm of the previous expression and set the first derivative equal to 0 to obtain

$$r_{\text{opt}} = -\log\left[\frac{p \cdot (1 + \mathtt{T} \cdot R)}{(1 + \mathtt{T} \cdot \mathtt{R} - \mathtt{T}) \cdot (-1 + p)}\right].$$

Fig. 9-4 plots the optimized bounds. Specifically, the right tail of each curve in the plot represents an upper bound on $\Pr[t > \mathtt{T}]$ while the left tail represents an upper bound on $\Pr[t < \mathtt{T}]$. Thus we see that if we transmit at rates approaching capacity, then with high probability, large decoding delays will be incurred. If we require a bound on the probability that a packet requires too large a delay, then Fig. 9-4 illustrates the corresponding maximum rate.

## ■ 9.6 Concluding Remarks

In this chapter we explored the decoding delay for general channel dynamics instead of the specific burst dynamics considered in Chapter 7. We introduced the idea of the achievable delay region and proved a coding theorem characterizing the delay region under certain conditions. Motivated by this theoretical analysis we developed a practical code construction for erasure channels with optimal delay properties. Finally, we explored what rates a channel can support if the maximum or expected delay is bounded.

Chapter 10

# Low Delay Application and Physical Layer Diversity Architectures

## ■ 10.1 Introduction

Consider transmitting a source such as audio, video, or speech over a wireless link. Due to the nature of wireless channels, effects such as fading, shadowing, interference from other transmitters, and network congestion can cause the channel quality to fluctuate during transmission. When the channel varies on a time-scale longer than the delay constraints of the desired application, such channel fluctuations cause outages. Specifically, when the channel quality is too low, the receiver will be unable to decode the transmitted data in time to reconstruct it at the appropriate point in the source stream. Thus some frames of video or segments of speech/audio will be reconstructed at the receiver with large distortions.

As illustrated in Fig. 10-1, one approach to combat such channel fluctuations is to code over multiple parallel channels (*e.g.*, different frequency bands, antennas, or time slots) and leverage diversity in the channel. A variety of source and channel coding schemes can applied to this scenario, including progressive and multiple description source codes [57,148,173,10,17,47,68,154,167,179,4,12,9,14,48,76,75,89,91,147,174,180,194,202,172, 201,3,71,184,203], broadcast channel codes [34,105,106,161,43,46], and hybrid analog-digital codes [38, Chapter 3] [36,37,155,127]; however, the best source and channel coding architecture to exploit such parallel channels is still unknown. In this chapter, we examine system architectures based upon two encoding algorithms that exploit diversity in the source coding and channel coding, respectively, along with two compatible decoding algorithm for the first encoder, and one compatible decoding algorithms for the second encoder. We compare performance of these systems by studying their average distortion performance on a various block fading channel models.

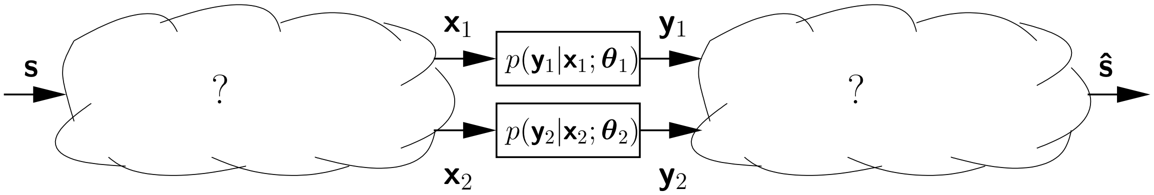

s
?
$\mathbf{x}_1$
$\mathbf{x}_2$
$p(\mathbf{y}_1|\mathbf{x}_1;\boldsymbol{\theta}_1)$
$p(\mathbf{y}_2|\mathbf{x}_2;\boldsymbol{\theta}_2)$
$\mathbf{y}_1$
$\mathbf{y}_2$
?
$\hat{\mathbf{s}}$

**Figure 10-1.** Conceptual illustration of the parallel diversity coding problem considered in this chapter. An encoder must map a source sequence, $\mathbf{s}$, into a pair of channel inputs $\mathbf{x}_1$ and $\mathbf{x}_2$ *without* knowing the channel states $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. A decoder must map the channel outputs $\mathbf{y}_1$ and $\mathbf{y}_2$ along with knowledge of the channel states into an estimate of the source, $\hat{\mathbf{s}}$. The optimal encoding and decoding architecture is unknown.

More specifically, Fig. 10-2 illustrates the two classes of encoders we consider. In the *channel coding diversity* system of Fig. 10-2(a), the source $s$ is encoded into $\hat{s}$ by a single description (SD) source coder. Next $\hat{s}$ is jointly encoded into $(x_1, x_2)$ by the channel coder and transmitted across a parallel channel. For the *source coding diversity* system of Fig. 10-2(b), the source $s$ is encoded into $\hat{s}_1$ and $\hat{s}_2$ by a multiple description (MD) source coder. Each $\hat{s}_i$ is then separately encoded into $x_i$ by a channel coder and transmitted across the appropriate channel.

Since the encoders in Fig. 10-2 exploit the inherent diversity of a parallel channel in qualitatively different ways, we focus on the following two questions:

1. Which of the basic architectures in Fig. 10-2 achieves the smallest average distortion? If neither architecture is universally best, for what channels is one architecture better than the other?

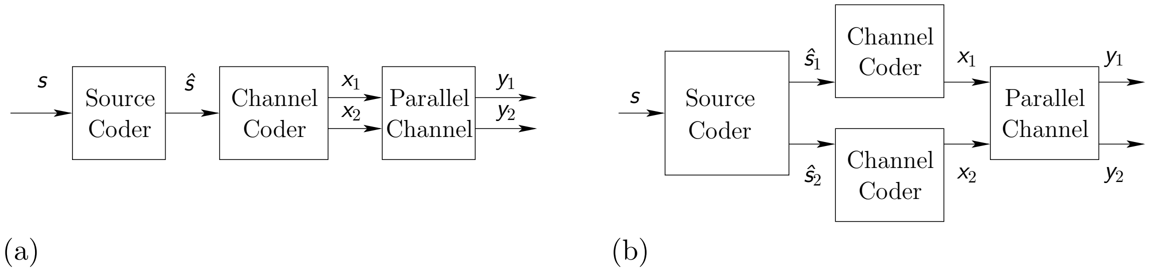


**Figure 10-2.** Diagrams for (a) channel coding diversity and (b) source coding diversity.

2. Is there a way to combine the best features of both systems in Fig. 10-2?

Essentially, the answers we develop can be illustrated through Fig. 10-3. For channel coding diversity, the source codeword, $\hat{s}$, can be reliably decoded only if the *total* channel quality is high enough to support the transmission rate. So this system achieves diversity in the sense that even if one of the channels is bad, then as long as the overall channel quality is good, the receiver will still be able to recover the encoded source. In contrast, for source coding diversity, each source codeword $\hat{s}_i$ can be decoded if the quality of the corresponding *individual* channel is high enough. This system achieves diversity in the sense that even if one of the channels is bad and one description is unrecoverable, then as long as the other channel is good and the remaining description is recovered, a low fidelity source reconstruction is obtained. If both channels are good and both descriptions are successfully decoded, then they are combined to form a high fidelity reconstruction.

Fig. 10-3 compares the two systems when the source coders are designed to achieve the same distortion if all source codewords are successfully decoded (*i.e.*, in region III). Furthermore, in region I, both systems fail to decode and again have the same distortion. In regions II and V, channel coding diversity is superior since the channel conditions are such that at most one source codeword is decoded under source coding diversity. Conversely, in region IV, source coding diversity is superior since one source codeword is received, and channel coding diversity fails to decode. Therefore our first question about which of the architectures in Fig. 10-2 is best, is essentially a question about which region the channel quality is most likely to lie in. If regions II and V are more probable, channel coding diversity will be superior; conversely, if regions IV are more likely, source coding diversity will be superior.

As a specific example, in the classic MD coding problem modeling link failure or packet erasure [71], each channel is either off, in which case no information can be communicated, or supports a particular rate. The four channel conditions for this scenario are indicated by ♦'s in Fig. 10-3 for an example packet erasure channel. For such discrete models, source coding diversity is clearly superior, since both SD and MD source coding achieve the same distortions in regions I and III, but channel coding diversity fails completely in region IV. In this region, source coding diversity recovers one source codeword and produces a low fidelity reconstruction of the source.

The opposite occurs for channels where a continuous range of rates can potentially be supported (*e.g.*, additive white Gaussian noise channels with Rayleigh fading). For these channels, the channel quality is essentially more likely to lie in region II than in IV and thus channel coding diversity is superior. Specifically, we characterize performance by analyzing how quickly the average distortion decays as a function of the signal-to-noise ratio (SNR)

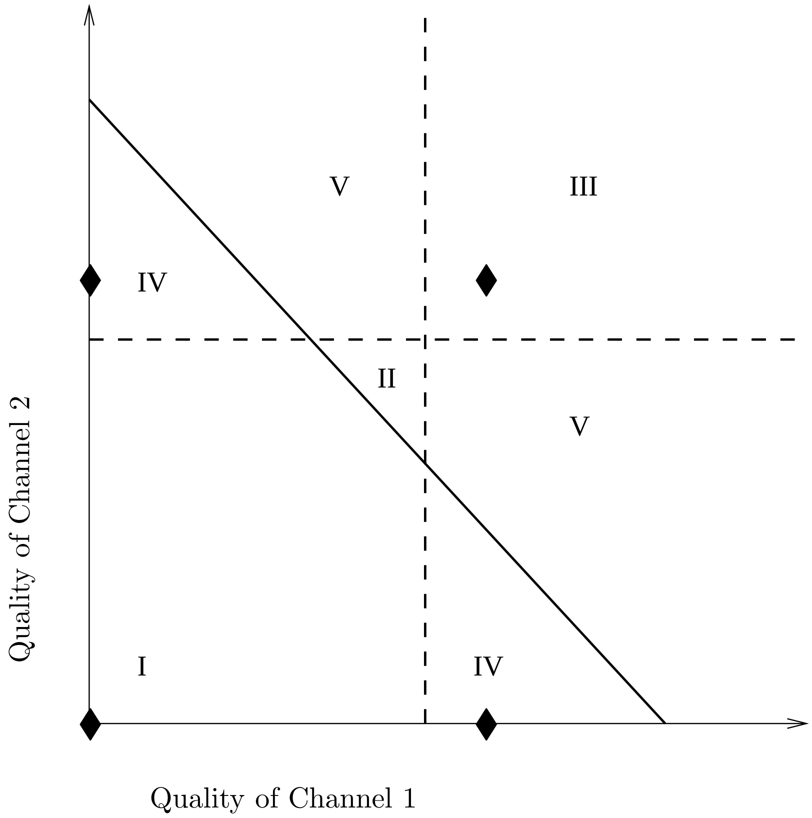


**Figure 10-3.** Conceptual illustration of successful decoding regions for source and channel coding diversity systems designed to have the same distortion when all codewords are received. For channel coding diversity, the receiver will be able to decode the transmitted source description if the sum of the channel quality exceeds a threshold represented by the solid diagonal line. For source coding diversity, the first (respectively, second) source description will be successfully decoded provided the first (resp., second) channel quality exceeds the vertical (resp., horizontal) dashed line. The ♦'s represent the four possible channel qualities for a packet loss channel where each channel is either on or off.

for various systems. We refer to the slope of the distortion versus SNR on a log-log plot as the "distortion exponent" and use this as our figure of merit. In particular, our analysis shows that optimal channel coding diversity is generally superior to source coding diversity on continuous channels in the sense that an optimal channel coding diversity architecture achieves a better distortion exponent than a source coding diversity architecture.

Since source coding diversity is best for on-off channels, and optimal channel coding diversity is best for continuous state channels, our second question of whether there exists an architecture that combines the advantages of both becomes relevant. In addition to our analysis of the two previously known diversity architectures in Fig. 10-2, our second main contribution is the description of a new joint source-channel decoding architecture which achieves the best qualities of both. Specifically, to perform well on both continuous state channels and on-off channels we do not propose a third *encoding* architecture, but a third new *joint decoding* architecture. We show that the main inefficiency of source coding diversity on continuous state channels results from the channel decoders ignoring the correlation between the multiple descriptions. By explicitly accounting for the structure of the source encoding when performing channel decoding, we prove a coding theorem characterizing the performance of source coding diversity with joint decoding. We show that such a system can achieve the same performance as optimal channel coding diversity on continuous channels and the same performance as source coding diversity for on-off channels.

### ■ 10.1.1 Related Research

The problem of MD coding was initially studied from a rate-distortion perspective, having been formalized by Gersho, Witsenhausen, Wolf, Wyner, Ziv, and Ozarow at the 1979 IEEE Information Theory Workshop. Their initial contributions to the problem appear in [183,185,140,184]. El Gamal & Cover develop an achievable rate region for two descriptions in [71], and this region is shown to be optimal for the Gaussian source, with mean-square distortion, by Ozarow [140]. Specialized results for the binary symmetric source, with Hamming distortion, are developed by Berger & Zhang [20, 201, 202] and Ahlswede [3]. Zamir [194] develops high-rate bounds for memoryless sources. Most recently, work by Venkatarami *et. al* [174, 173] provides achievable rate regions for many descriptions that generalize the results in [71, 201]. Important special cases of the MD coding problem have also been examined, including successive refinement, or layered coding, [57, 156] and certain symmetric cases [147, 148].

Some practical approaches to MD coding include MD scalar quantization, dithered MD lattice quantization, and MD transform coding. Vaishampayan [172] pioneered the former, Frank-Dayan and Zamir considered the use of dither [68], and Wang, Orchard, Vaishampayan, and Reibman [180] and later Goyal & Kovacevic [76] studied the latter. See [75] for a thorough review of both approaches. Recently, the design of MD video coders has received considerable attention [9, 10, 154, 91, 167, 179]

All of the classical work on MD coding utilizes an "on-off" model for the channels or networks under consideration, without imposing strict delay constraints. More specifically, source codes are designed assuming that each description is completely available (error-free) at the receiver, or otherwise completely lost. Furthermore, the likelihood of these events occurring is independent of the choice of source coding rates. Under such conditions, it is not surprising that MD coding outperforms SD coding; however, for many practical channel and network environments, these conditions do not hold. For example, in delay constrained situations, suitable for real-time or interactive communication, descriptions may have to be encoded as multiple packets, each of which might be received or lost individually. Furthermore, congestion and outage conditions often depend heavily upon the transmission rate. Thus, it is important to consider MD coding over more practical channel models, as well as to fairly compare performance with SD coding.

Some scattered work is appearing in this area. Ephremides *et. al* [4] examine MD coding over a parallel queue channel, compare to SD coding, and show that MD coding offers significant advantages under high traffic (congestion) situations. This essentially results because the MD packets are more compact than SD packets, and indicates the importance of considering the influence of rate on congestion. Coward *et. al* [48, 47] examine MD coding over several channel models, including memoryless symbol-erasure and symbol-error channels, as well as block fading channels. For strict delay constraints, they show that MD outperforms SD; for longer delay constraints, allowing for more sophisticated channel coding, they show that SD outperforms MD. Thus, the impact of delay constraints are important. This chapter examines fading conditions similar to those in [48,47], but considers a wider variety of channel coding and decoding options, with an emphasis on architectural considerations as well as performance.

### ■ 10.1.2 Outline

We begin by summarizing our system model in Section 10.2. Section 10.3 studies on-off channels, Section 10.4 treats continuous state channels, and Section 10.5 develops source coding diversity with joint decoding. Many of the more detailed proofs are deferred to Appendices. Finally, Section 10.6 closes the chapter with some concluding remarks and directions for further research.

## ■ 10.2 System Model

Fig. 10-1 depicts the general system model we consider in this chapter. Our objective is to design and evaluate methods for communicating a source signal $\mathsf{s}$ with small distortion over certain channels with independent parallel components. In particular, focusing on memoryless source models for simplicity of exposition, we consider non-ergodic channels models in which delay constraints or limited channel variations limit the effective blocklength at the encoder. Of many possible examples, we focus on on-off channels and additive noise channels with block fading. While cross-layer design is generally acknowledged to yield superior performance to layered design, simultaneously optimizing all facets of a system is usually too complex. Hence we consider various architectures based upon using a classical system at one layer combined with an optimized system at another layer. In the remainder of this section, after briefly introducing some notation, we summarize the source and channel models, discuss architectural options for encoding and decoding, and review high-resolutions approximations for the various source coding algorithms employed throughout the chapter.

### ■ 10.2.1 Notation

Vectors and sequences are denoted in bold (*e.g.*, $\mathbf{x}$) with the $i$th element denoted as $x[i]$. Random variables are denoted using the sans serif font (*e.g.*, $\mathsf{x}$) while random vectors and sequences are denoted with bold sans serif (*e.g.*, $\boldsymbol{\mathsf{x}}$). We denote mutual information, differential entropy, and expectation as $I(\mathsf{x};\mathsf{y})$, $h(\mathsf{x})$, $E[\mathsf{x}]$. Calligraphic letters denote sets (*e.g.*, $s \in \mathcal{S}$). When its argument is a set or alphabet, $|\cdot|$ denotes the cardinality of the argument. To simplify the discussion of architectures, we use the symbols $\mathrm{ENC}(\cdot)$ and $\mathrm{DEC}(\cdot)$ to denote a generic encoder and decoder. To specialize this generic notation to one of the architectures discussed in Section 10.2.4, we will employ subscripts representing the relevant system variables.

## ■ 10.2.2 Source Model

We model the source as a sequence of independent and identically distributed (i.i.d.) samples $\mathsf{s}[k]$. For example, such a discrete-time source may be obtained from sampling a continuous-time, appropriately band-limited, white-noise random process. We denote the probability density for the discrete-time source sequence $\mathsf{s}[k]$ as

$$p_{\mathbf{s}}(\mathbf{s}) = \prod_{k=1}^{K} p_{\mathsf{s}}(s[k]) \ . \tag{10.1}$$

We assume that the process is such that the differential entropy, $h(\mathsf{s})$, and second moment, $\mathrm{E}\left[\mathsf{s}^2\right]$, both exist and are finite.

To measure quality of the communication system, we employ a distortion measure between the source signal $\mathsf{s}$ and its reconstruction $\hat{\mathsf{s}} \in \hat{\mathcal{S}}$. Specifically, given a per-letter distortion measure $d(\mathsf{s}[k], \hat{\mathsf{s}}[k])$, we extend it additively to blocks of source samples, *i.e.*,

$$d(\mathbf{s}, \hat{\mathbf{s}}) = \sum_{k=1}^{K} d(\mathsf{s}[k], \hat{\mathsf{s}}[k]) \ . \tag{10.2}$$

We may characterize performance in terms of various statistics of the distortion, viewed as a random variable. In particular, we focus on the expected distortion

$$D = \mathrm{E}\left[d(\mathbf{s}, \hat{\mathbf{s}})\right] \ . \tag{10.3}$$

Throughout our development, we will emphasize squared-error distortion, for which $d(\mathsf{s}, \hat{\mathsf{s}}) = (\mathsf{s} - \hat{\mathsf{s}})^2$; in this case, (10.3) is the mean-square distortion.

## ■ 10.2.3 (Parallel) Channel Model

The channel depicted by Fig. 10-1 consists of two branches, each of which corresponds to an independent channel with independent states. Specifically, a channel input block, $\mathbf{x}$, consists of two sub-blocks, $\mathbf{x}_1$ and $\mathbf{x}_2$, and the corresponding channel output block, $\mathbf{y}$, consists of the two sub-blocks, $\mathbf{y}_1$ and $\mathbf{y}_2$. The channel states are denoted by random variables $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, respectively. The channel law is the product of the two independent sub-channel laws:

$$\begin{aligned} p_{\mathbf{y}_1,\mathbf{y}_2,\boldsymbol{\theta}_1,\boldsymbol{\theta}_2|\mathbf{x}_1,\mathbf{x}_2}(\mathrm{y}_1, \mathrm{y}_2, \theta_1, \theta_2 | \mathrm{x}_1, \mathrm{x}_2) = p_{\mathbf{y},\boldsymbol{\theta}|\mathbf{x}}(\mathrm{y}_1, \theta_1|\mathrm{x}_1) \cdot p_{\mathbf{y},\boldsymbol{\theta}|\mathbf{x}}(\mathrm{y}_2, \theta_2|\mathrm{x}_2) = \\ p_{\boldsymbol{\theta}}(\theta_1) \cdot p_{\boldsymbol{\theta}}(\theta_2) \prod_{i=1}^{n_c} \left[p_{\mathsf{y}|\mathsf{x},\boldsymbol{\theta}}(y_1[i]|x_1[i], \theta_1) \cdot p_{\mathsf{y}|\mathsf{x},\boldsymbol{\theta}}(y_2[i]|x_2[i], \theta_2)\right] . \end{aligned} \tag{10.4}$$

For simplicity, we only consider channels for which the input distribution that maximizes the mutual information is independent of the channel state. Throughout the chapter we consider the case where both the transmitter and receiver know the channel state distribution $p_{\boldsymbol{\theta}}$ and the channel law $p_{y|x}$, but only the receiver knows the realized channel states and channel outputs.

To examine fundamental performance and compare between systems, we analyze random coding over these non-ergodic channels using outage probability [142] as a performance measure. Briefly, because the mutual information $\mathsf{I}$, corresponding to the supportable transmission rate of the channel, is a function of the fading coefficients or other channel uncertainty, it too is a random variable. For fixed transmission rate $R$ (in nats/channel use), the outage probability $\Pr[\mathsf{I} < R]$ measures channel coding robustness to uncertainty in the channel.

The structure of the channel coding and decoding affects the form of the outage probability expression [142]. If coding is performed over only the first component channel, then the probability of decoding failure is $\Pr[\mathsf{I}(\mathbf{x}_1; \mathbf{y}_1) < R]$. If repetition coding is performed across the parallel channels, then a single message is encoded as $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}$. With selection combining at the receiver, the probability of decoding failure is $\Pr\{\max[\mathsf{I}(\mathbf{x}; \mathbf{y}_1), \mathsf{I}(\mathbf{x}; \mathbf{y}_2)] < R\}$; with optimal maximum-ratio combining at the receiver, the probability of decoding failure is $\Pr\{\mathsf{I}(\mathbf{x}; \mathbf{y}_1, \mathbf{y}_2) < R\}$. Finally, if optimal parallel channel coding is performed using a pair of jointly-designed codebooks with $\mathbf{x}_1$ and $\mathbf{x}_2$ independent, the probability of decoding failure is $\Pr[\mathsf{I}(\mathbf{x}_1; \mathbf{y}_1) + \mathsf{I}(\mathbf{x}_2; \mathbf{y}_2) < R]$.

## ■ 10.2.4 Architectural Options

In this section, we specify some architectural options for encoding and decoding in the source-channel diversity system depicted in Fig. 10-1.

### Joint Source-Channel Diversity

In the most general setup, joint source-channel diversity consists of a pair of mappings $(\mathrm{ENC}_{\mathbf{x}_1,\mathbf{x}_2 \leftarrow \mathbf{s}}, \mathrm{DEC}_{\hat{\mathbf{s}} \leftarrow \mathbf{y}_1,\mathbf{y}_2})$. The encoder $\mathrm{ENC}_{\mathbf{x}_1,\mathbf{x}_2 \leftarrow \mathbf{s}}$ maps a sequence of $K$ source letters into $N$ pairs of channel inputs; correspondingly, the decoder maps $N$ pairs of channel outputs into $K$ reconstruction letters. The ratio $N/K$ (sometimes referred to as the processing gain, excess bandwidth, or bandwidth expansion factor) is denoted with the symbol $\beta \triangleq N/K$.[1] Mathematically,

$$\mathrm{ENC}_{\mathbf{x}_1,\mathbf{x}_2 \leftarrow \mathbf{s}} : \mathcal{S}^K \longrightarrow \mathcal{X}_1^N \times \mathcal{X}_2^N \tag{10.5}$$

$$\mathrm{DEC}_{\hat{\mathbf{s}} \leftarrow \mathbf{y}_1,\mathbf{y}_2} : \mathcal{Y}_1^N \times \mathcal{Y}_2^N \longrightarrow \hat{\mathcal{S}}^K \ . \tag{10.6}$$

If the image of $\mathrm{ENC}_{\mathbf{x}_1,\mathbf{x}_2 \leftarrow \mathbf{s}}$, *i.e.*, $\mathrm{ENC}_{\mathbf{x}_1,\mathbf{x}_2 \leftarrow \mathbf{s}}(\mathcal{S}^K)$, is finite, we define the *rate* of the code as

$$R = \frac{\ln |\mathrm{ENC}_{\mathbf{x}_1,\mathbf{x}_2 \leftarrow \mathbf{s}}(\mathcal{S}^K)|}{N} \ , \tag{10.7}$$

which has units of nats per parallel channel use.

Regarding the non-ergodic nature of the channels, we consider situations in which $K$ is large enough to average over source fluctuations, *i.e.*, the source is ergodic, but $N$ is not large enough to average over channel variations, *i.e.*, the channel is non-ergodic.

[1] The bandwidth expansion ratio in [96] (denoted by $L$) is defined slightly differently from $\beta$. Specifically, since [96] considers a complex source and Rayleigh fading Gaussian noise channel, $L = 2\beta$.

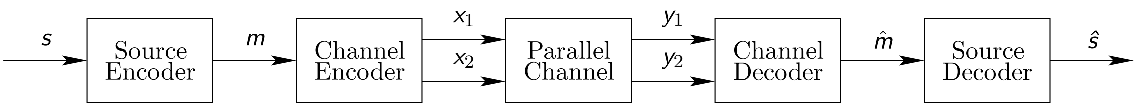


**Figure 10-4.** Channel coding diversity.

### Channel Coding Diversity

From one perspective, a natural way to exploit diversity in the channel is to employ repetition or more powerful channel codes applied to a single digital representation of the source. In such scenarios, Fig. 10-1 specializes to that shown in Fig. 10-4. Such channel coding diversity consists of a source pair of encoder and decoder mappings $(\text{ENC}_{m\leftarrow\mathbf{s}}, \text{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}})$ and a channel pair of encoder and decoder mappings $(\text{ENC}_{\mathbf{x}\leftarrow m}, \text{DEC}_{\hat{m}\leftarrow\mathbf{y}})$. As in classical rate-distortion source coding, the source encoder maps a sequence of $K$ input letters to a finite index, and the source decoder maps an index into a sequence of $K$ reconstruction letters:

$$\text{ENC}_{m\leftarrow\mathbf{s}} : \mathcal{S}^K \longrightarrow \{1, 2, \ldots, |\mathcal{M}|\} \tag{10.8}$$

$$\text{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}} : \{0, 1, 2, \ldots, |\mathcal{M}|\} \longrightarrow \hat{\mathcal{S}}^K \tag{10.9}$$

Further, as in classical channel coding, the channel encoder maps an index into $N$ pairs of channel inputs, and the channel decoder maps $N$ pairs of channel outputs into an index:

$$\text{ENC}_{\mathbf{x}\leftarrow m} : \{1, 2, \ldots, |\mathcal{M}|\} \longrightarrow \mathcal{X}_1^N \times \mathcal{X}_2^N \tag{10.10}$$

$$\text{DEC}_{\hat{m}\leftarrow\mathbf{y}} : \mathcal{Y}_1^N \times \mathcal{Y}_2^N \longrightarrow \{0, 1, \ldots, |\mathcal{M}|\} \; . \tag{10.11}$$

Note that we include the index 0 at the output of the channel decoder and input to the source decoder. This serves as a flag in the event of a (detected) channel coding error.

For the channel coding diversity approach, a key parameter is the rate defined by

$$R = \frac{\ln|\mathcal{M}|}{N} \; , \tag{10.12}$$

where again the units are nats per parallel channel use.

### Source Coding Diversity

Instead of exploiting diversity through channel coding, an emerging class of source coding algorithms based upon MD coding allows diversity to be exploited by the source coding layer.

For such source coding diversity, the block diagram of Fig. 10-1 specializes to that shown in Fig. 10-5. Source coding diversity employs two independent, but otherwise classical, channel encoder and decoder pairs $(\text{ENC}_{\mathbf{x}_1\leftarrow m_1}, \text{DEC}_{\hat{m}_1\leftarrow\mathbf{y}_1})$ and $(\text{ENC}_{\mathbf{x}_2\leftarrow m_2}, \text{DEC}_{\hat{m}_2\leftarrow\mathbf{y}_2})$:

$$\text{ENC}_{\mathbf{x}_i\leftarrow m_i} : \{1, 2, \ldots, |\mathcal{M}_i|\} \longrightarrow \mathcal{X}_i^N \tag{10.13}$$

$$\text{DEC}_{\hat{m}_i\leftarrow\mathbf{y}_i} : \mathcal{Y}_i^N \longrightarrow \{0, 1, 2, \ldots, |\mathcal{M}_i|\} \; , \tag{10.14}$$

for $i = 1, 2$. Again, we allow for the output of the channel decoding process to be 0 to

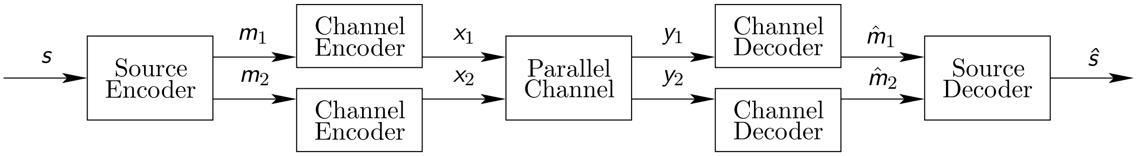


**Figure 10-5.** Source coding diversity system model described more precisely in Section 10.2.4.

| $\hat{m}_1$ | $\hat{m}_2$ | $\mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}}$ |
|---|---|---|
| $=0$ | $=0$ | $\mathrm{DEC}_{\hat{\mathbf{s}}_\emptyset\leftarrow\hat{m}_\emptyset}$ |
| $=0$ | $\neq 0$ | $\mathrm{DEC}_{\hat{\mathbf{s}}_1\leftarrow\hat{m}_1}$ |
| $\neq 0$ | $=0$ | $\mathrm{DEC}_{\hat{\mathbf{s}}_2\leftarrow\hat{m}_2}$ |
| $\neq 0$ | $\neq 0$ | $\mathrm{DEC}_{\hat{\mathbf{s}}_{1,2}\leftarrow\hat{m}_{1,2}}$ |

**Table 10.1.** Source coding diversity decoder rules based upon channel conditions.

indicated a (detected) error. Here the rates

$$R_i = \frac{\ln|\mathcal{M}_i|}{N} \ , \quad i = 1,2 \ , \tag{10.15}$$

both in nats per parallel channel use, are key parameters of the system.

The source encoder consists of two mappings

$$\mathrm{ENC}_{m_i\leftarrow\mathbf{s}} : \mathcal{S}^K \longrightarrow \{1,2,\ldots,|\mathcal{M}_i|\} \ , \quad i = 1,2 \ . \tag{10.16}$$

The source decoder can be viewed as four separate mappings, depending upon whether or not there are channel decoding errors on the individual channels. Specifically, the source decoder can be constructed from the following four mappings:

$$\mathrm{DEC}_{\hat{\mathbf{s}}_\emptyset\leftarrow\hat{m}_\emptyset} : \{0\}\times\{0\} \longrightarrow \{s_*\}^K \tag{10.17}$$

$$\mathrm{DEC}_{\hat{\mathbf{s}}_1\leftarrow\hat{m}_1} : \{1,2,\ldots,|\mathcal{M}_1|\}\times\{0\} \longrightarrow \hat{\mathcal{S}}_1^K \tag{10.18}$$

$$\mathrm{DEC}_{\hat{\mathbf{s}}_2\leftarrow\hat{m}_2} : \{0\}\times\{1,2,\ldots,|\mathcal{M}_2|\} \longrightarrow \hat{\mathcal{S}}_2^K \tag{10.19}$$

$$\mathrm{DEC}_{\hat{\mathbf{s}}_{1,2}\leftarrow\hat{m}_{1,2}} : \{1,2,\ldots,|\mathcal{M}_1|\}\times\{1,2,\ldots,|\mathcal{M}_2|\} \longrightarrow \hat{\mathcal{S}}_0^K \ , \tag{10.20}$$

where $s_*$ is a constant determined by the distortion measure for the source; for example, if mean-square distortion is important, then $s_* = \mathrm{E}[s]$. Table 10.1 summarizes how these mappings are employed.

### Source Coding Diversity with Joint Decoding

Finally, we also consider source coding diversity with joint decoding, as depicted in Fig. 10-6. Here all is the same as in the source coding diversity model of Fig. 10-5, except that source and channel decoding is performed jointly across channels by accounting for correlation among the channel coding inputs $m_1$ and $m_2$. Specifically, the channel decoding for this approach is a mapping

$$\mathrm{DEC}_{\hat{m}_{1,2}\leftarrow\mathbf{y}_{1,2}} : \mathcal{Y}_1^N\times\mathcal{Y}_2^N \longrightarrow \{0,1,2,\ldots,|\mathcal{M}_1|\}\times\{0,1,2,\ldots,|\mathcal{M}_1|\} \tag{10.21}$$

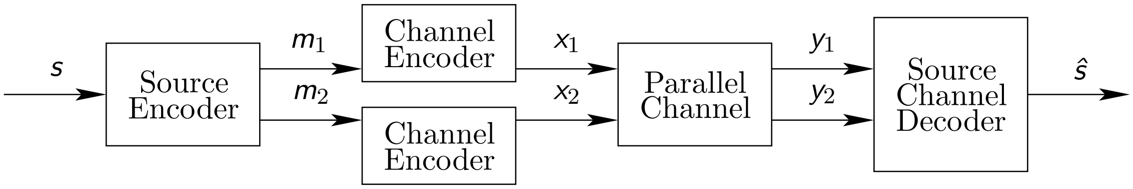


**Figure 10-6.** Source coding diversity with joint source-channel decoding.

which also takes into account knowledge of the source coding structure. In practice full joint-design of the decoder may not be required and a partially separated design where likelihood-ratios, quantized likelihood-ratios or similar information are exchanged between the source and channel decoders may be sufficient.

## 10.2.5 High-Resolution Approximations for Source Coding

An important practical example of our source model is the Gaussian source, for which $p_{\mathsf{s}}(s)$ is a Gaussian density function with zero mean and unit variance. The Gaussian source also serves as a useful approximation to other sources in the high resolution (low distortion) regime [109, 194]. We now summarize the well-known results for single- and multiple-description source coding for the Gaussian case, and generalize them using the high resolution distortion approximations. These high resolution approximations are utilized throughout the sequel in our performance analysis.

### Single Description Source Coding

In SD source coding, or classical rate-distortion theory, the source, $\mathsf{s}$, is quantized into a single description, $\hat{\mathsf{s}}$, using rate $R$.

In general, the rate-distortion function is difficult to determine, but a number of researchers have determined the rate-distortion function in the high resolution limit. Specifically, under some mild technical conditions [109],

$$\lim_{D\to 0} R(D) - \frac{1}{2}\log\frac{e^{2h(\mathsf{s})}}{2\pi e D} = 0\ . \tag{10.22}$$

This result also implies that[2]

$$R(D) \approx \frac{1}{2}\log\frac{e^{2h(\mathsf{s})}}{2\pi e D} \tag{10.23}$$

Without loss of generality we scale a given source under consideration so that $e^{2h(\mathsf{s})} = 2\pi e$ to simplify the notation. Furthermore, instead of measuring the quantization rate in bits, we will find it more convenient to measure the rate in nats per channel sample by using the processing gain $\beta$ defined in Section 10.2.4. Thus we will use the expressions

$$R(D) \approx \frac{1}{2\beta}\ln\frac{1}{D} \quad \text{and} \quad \exp R(D) \approx D^{-1/(2\beta)} \tag{10.24}$$

[2] Throughout the chapter, the approximation $f(x) \approx g(x)$ is in the sense that $f(x)/g(x) \to 1$ and $|f(x) - g(x)| \to 0$ as $x$ approaches a limit, either $x \to 0$ or $x \to \infty$, which should be clear from the context.

to approximate $R(D)$ and $\exp R(D)$ in high-resolution.

As is well-known, the rate (in nats/channel sample) required for SD source coding of a Gaussian source at average distortion $D$ for any resolution is [46]

$$R_{\mathrm{sd}}(D) = \frac{1}{2\beta} \log \frac{1}{D} \ . \tag{10.25}$$

Therefore, one way to interpret (10.23), is that for difference distortion measures in the high-resolution limit all sources essentially look Gaussian except for scaling by the constant factor $\exp[2h(\mathsf{s})]/(2\pi e)$. Note that the form of the rate-distortion function in (10.23) is asymptotically accurate and not a worst case result like those in [100, 159].

### Multiple Description Source Coding

In contrast to SD coding, MD source coding quantizes the source into two descriptions, $\hat{\mathbf{s}}_1$ and $\hat{\mathbf{s}}_2$ so that if only one is received then moderate distortion is incurred, and if both descriptions are received then lower distortion is obtained [71].

The rates and distortions achievable by coding a unit variance Gaussian source into two equal-rate descriptions with a total rate of $R_{\mathrm{md}}$ nats per channel sample, (*i.e.*, each description requires $R_{\mathrm{md}}/2$ nats) satisfy [71]

$$R_{\mathrm{md}}(D_0, D_1) = \frac{1}{2\beta} \log \frac{1}{D_0} + \frac{1}{2\beta} \log \frac{(1-D_0)^2}{(1-D_0)^2 - (1-2D_1+D_0)^2} \ , \tag{10.26a}$$

in the case of low distortions ($2D_1 - D_0 \leq 1$) where $D_0$ is the distortion when both descriptions are received and $D_1$ is the description when only a single description is received. For high distortions with ($2D_1 - D_0 \geq 1$), there is no penalty for the multiple descriptions and the total rate required is

$$R_{\mathrm{md}}(D_0, D_1) = \frac{1}{2\beta} \log \frac{1}{D_0}. \tag{10.26b}$$

The general rate-distortion region for the MD coding problem is still unknown, in the Gaussian case for more than two descriptions, and for more general sources. In the high resolution limit the rate-distortion region is the same as for a Gaussian source with variance $\exp[2h(\mathsf{s})]/(2\pi e)$ [194]. Hence for our asymptotic analysis we use the rate distortion function in (10.26) for both Gaussian and non-Gausian sources with $\exp[2h(\mathsf{s})]/(2\pi e) = 1$.

Exponentiating (10.26a) yields

$$\begin{aligned}
\exp[R_{\mathrm{md}}(D_0, D_1)] &= D_0^{-1/(2\beta)} \cdot (1-D_0)^{-1/\beta} \\
&\quad \cdot (1 - 2D_0 + D_0^2 - 1 - D_0^2 - 2D_0 + 4D_1 + 4D_0D_1 - 4D_1^2)^{-1/(2\beta)} \qquad (10.27)\\
&= D_0^{-1/(2\beta)} \cdot (1-D_0)^{-1/(\beta)} \cdot (4D_1 - 4D_0 + 4D_0D_1 - 4D_1^2)^{-1/(2\beta)} \qquad (10.28)\\
&\approx D_0^{-1/(2\beta)} \cdot (4D_1 - 4D_0)^{-1/(2\beta)} \qquad (10.29)
\end{aligned}$$

where the last line follows since $(1 - D_0) \approx 1$ and $4(D_1 - D_0 + D_0D_1 - D_1^2) \approx 4(D_1 - D_0)$ as $D_0 \to 0$ and $D_1 \to 0$. If only $D_0 \to 0$, then the $\approx$ in (10.29) must be replaced with $\gtrapprox$. Any reasonable multiple description system has $D_0 \leq D_1/2$ (otherwise the denominator of (10.26a) could be easily increased while decreasing the distortion by setting $D_1 = 2D_0$). So since $2D_1 \leq 4(D_1 - D_0) \leq 4D_1$ we obtain

$$(4D_0 D_1)^{-1/(2\beta)} \lessapprox \exp[R_{\mathrm{md}}(D_0, D_1)] \lessapprox (2D_0 D_1)^{-1/(2\beta)} \qquad (10.30)$$

where the lower bound holds when $D_0 \to 0$ and the upper bound also requires $D_1 \to 0$.

## ■ 10.3 On-Off Component Channels

In this section, we examine the performance of source and channel coding diversity for scenarios in which each of the component channels is either "on", supporting a given transmission rate, or "off", supporting no rate (or an arbitrarily small rate). Much of the literature suggests that source coding diversity was developed for, and performs well on, such channel models. Our analysis is based upon channels that are parameterized in a manner similar to the continuous channels in Section 10.4. This parameterization allows us to compare source and channel coding diversity over a broad range of operating conditions. In addition to confirming that there exist operating conditions for which source coding diversity significantly outperforms channel coding diversity, our results illustrate that there also exist operating conditions for which the performance difference between source and channel coding diversity is negligible.

### ■ 10.3.1 Component Channel Model

For cases in which we are concerned with prolonged, deep fading or shadowing in a mobile radio channel, strong first-adjacent interference in a terrestrial broadcast channel, or congestion in a network, we can model the channel state $\boldsymbol{\theta}_i$ as taking on only two possible values. Specifically, we can consider on-off channels where the channel mutual information has probability law

$$I = \begin{cases} \ln(1 + \mathrm{SNR}) \ , & \text{with probability } (1 - \epsilon) \\ 0 \ , & \text{with probability } \epsilon \end{cases} . \qquad (10.31)$$

In (10.31), SNR parameterizes the channel quality when the channel is on, and $\epsilon$ parameterizes the probability that the channel is off. There is no connection between the channels' probability of being off and the quality in the on state; that is, neither SNR nor the selected encoding rate $R$ effects $\epsilon$. By contrast, for the continuous channels discussed in Section 10.4, $\epsilon$ will depend directly on both.

For simplicity of exposition, and ease of comparison with continuous channel scenarios in the sequel, the term *outage* will refer to the inability of a given approach to convey information over the pair of component channels. If both channels are off, then the system experiences outage regardless of the communication approach; however, as we will see, different approaches may or may not experience outage when one of the channels is on and the other is off. For all of the approaches we discuss, due to the nature of the on-off channels, performance can be classified into two regimes. The *quality-limited regime* has average distortion performance varying dramatically with the channel quality in the on state, because the distortion under no outage dominates the average distortion. In this case, the distortion under no outage is limited by the rate communicated, which, in turn, is limited by the channel quality. The *outage-limited regime* has average distortion performance that does not vary dramatically with the channel quality in the on state, because the distortion under outage dominates the average distortion.

### 10.3.2 No Diversity

Combining a SD source coder with a single component channel with channel encoder and decoder, the average distortion, as a function of the source coding rate $R$, is given by

$$\mathrm{E}\left[D_{\mathrm{NO-DIV}}(R)\right] = \begin{cases} (1-\epsilon)\exp(-2\beta) + \epsilon \ , & \text{if } 0 < R \leq \ln(1+\mathrm{SNR}) \\ 1 \ , & \text{otherwise} \end{cases} \ . \qquad (10.32)$$

Thus, the minimum average distortion is

$$\begin{aligned} D_{\mathrm{NO-DIV}} &= \min_{R} \mathrm{E}\left[D_{\mathrm{NO-DIV}}(R)\right] \\ &= (1-\epsilon)(1+\mathrm{SNR})^{-2\beta} + \epsilon \ . \end{aligned} \qquad (10.33)$$

We say that this system operates in the quality-limited regime if

$$(1+\mathrm{SNR})^{2\beta} \ll \frac{1-\epsilon}{\epsilon} \ , \qquad (10.34)$$

in which case, the average distortion behaves essentially as $(1-\epsilon)(1+\mathrm{SNR})^{-2\beta}$. If

$$(1+\mathrm{SNR})^{2\beta} \gg \frac{1-\epsilon}{\epsilon} \ , \qquad (10.35)$$

the system operates in the outage-limited regime, in which case the average distortion behaves essentially as $\epsilon$.

### 10.3.3 Optimal Channel Coding Diversity

Combining a SD source coder with optimal parallel channel coding over the component channels, the average distortion, as a function of the source coding rate $R$, is given by

$$\mathrm{E}\left[D_{\mathrm{OPT-CCDIV}}(R)\right] = \begin{cases} (1-\epsilon^2)\exp(-2\beta R) + \epsilon^2 \ , & \text{if } 0 < R \leq \ln(1+\mathrm{SNR}/2) \\ (1-\epsilon)^2\exp(-2\beta R) + [2\epsilon - \epsilon^2] \ , & \begin{array}{l} \text{if } \ln\left(1+\frac{\mathrm{SNR}}{2}\right) < R \\ \text{and } R \leq 2\ln\left(1+\frac{\mathrm{SNR}}{2}\right) \end{array} \\ 1 \ , & \text{otherwise} \end{cases} \ . \qquad (10.36)$$

For parallel channel coding, the two channel codewords are independent, and the system is able to sum the mutual informations of the component channels. This leads to the upper bound of $R \leq 2\ln(1+\mathrm{SNR}/2)$ in the second case of (10.36). If we instead utilized repetition coding, so that the two channel codewords are identical, the upper bound in the second case would instead be $R \leq \ln(1+\mathrm{SNR})$.

In contrast to the case of no diversity, the performance of the optimal channel coding diversity exhibits a discontinuity as a function of $R$. Fig. 10-7 illustrates that, because of the discrete probability distribution on the channel states, a discontinuity arises in the outage probability about the point $R = \ln(1+\mathrm{SNR}/2)$.

Clearly, each case in (10.36) is minimized by utilizing the largest possible rate for that

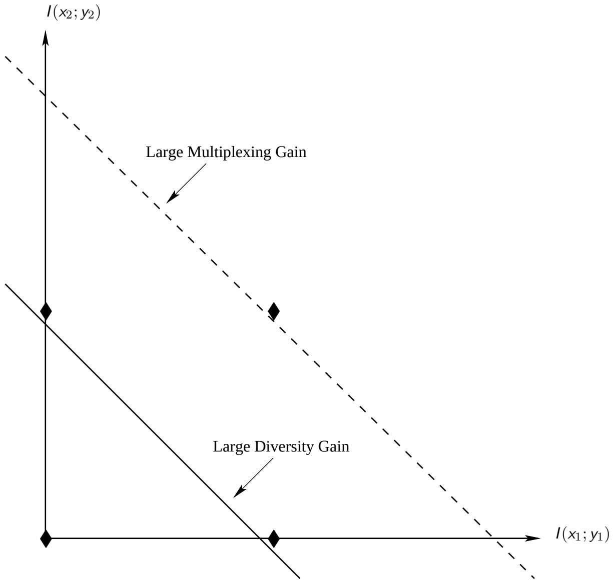


**Figure 10-7.** Outage region boundaries for optimal parallel channel coding. The ◆ symbols correspond to the sample mutual information pairs $(0,0)$, $(0, \ln(1{+}\mathrm{SNR}/2))$, $(\ln(1{+}\mathrm{SNR}/2), 0)$, and $(\ln(1{+}\mathrm{SNR}/2), \ln(1{+}\mathrm{SNR}/2))$. The solid line corresponds to the first case of (10.36), in which a low rate is selected to take advatange of diversity gain. The dashed line corresponds to the second case of (10.36), in which a higher rate is selected to take advantage of multiplexing gain. Outage regions are below and to the left of these diagonals.

case. Then the minimum average distortion becomes

$$\begin{aligned} D_{\mathrm{OPT-CCDIV}} &= \min_{R} \mathrm{E}\left[D_{\mathrm{OPT-CCDIV}}(R)\right] \\ &= \min \Big\{ (1-\epsilon^2)(1+\mathrm{SNR}/2)^{-2\beta} + \epsilon^2 \ , \\ &\qquad (1-\epsilon)^2(1+\mathrm{SNR}/2)^{-4\beta} + [1-(1-\epsilon)^2] \Big\} \ . \end{aligned} \tag{10.37}$$

As Fig. 10-8 illustrates, the two terms in (10.37) have their own quality- and outage-limited regimes, which, when combined by the minimum operation, leads to four trends in the overall system performance.

Comparing the two terms in (10.37), we see that the different choices of rate lead to different costs and benefits. Using the lower transmission rate, $R = \ln(1+\mathrm{SNR}/2)$, (*cf.* the first term in (10.37)) results in better outage-limited performance, but worse quality-limited performance. This approach exploits the diversity gain of the underlying parallel channel. On the other hand, using the higher transmission rate, $R = 2\ln(1+\mathrm{SNR}/2)$, (*cf.* the second term in (10.37)) results in worse outage-limited performance, but better quality-limited performance. This approach exploits the multiplexing gain of the underlying parallel channel. We note that the diversity and multiplexing terminology is inspired by the inherent tradeoff between the two for multiple-input, multiple-output (MIMO) wireless systems operating over fading channels [204].

Note that the two terms in (10.37) are equal when

$$(1+\mathrm{SNR}/2)^{2\beta} = \frac{1-\epsilon}{2\epsilon} \ . \tag{10.38}$$

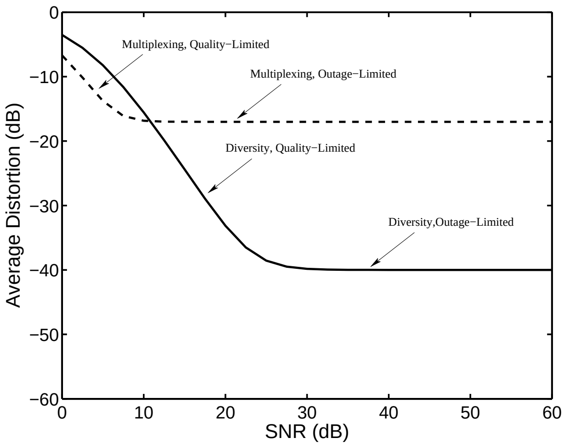


**Figure 10-8.** Average distortion performance with $\epsilon = 10^{-2}$ for the first (solid line) and second (dashed line) terms in the minimization of (10.37).

For small SNR (such that $(1+\text{SNR}/2)^{2\beta} < (1-\epsilon)/(2\epsilon)$), we exploit the multiplexing mode of operation and pass through its quality-limited and outage-limited regimes as we increase SNR until (10.38) is satisfied. As we will see, passing through the outage-limited regime of the multiplexing mode is the key limitation of optimal channel coding diversity for on-off channels. For higher SNR (such that $(1+\text{SNR}/2)^{2\beta} > (1-\epsilon)/(2\epsilon)$), we exploit the diversity mode of operation and pass through its quality- and outage-limited regimes as we increase SNR.

## ■ 10.3.4 Source Coding Diversity

In this section, we approximate the minimum average distortion for an MD system with independent channel coding. The analysis of this system is slightly more involved than those of previous sections because the rate-distortion region for MD coding is more complex, and independent channel coding over on-off component channels involves a pair of outage events.

Similar to Fig. 10-7, Fig. 10-9 displays outage region boundaries for independent channel coding. It is straightforward to see that the source coder should employ rates no greater than $\ln(1 + \text{SNR}/2)$ on each of the component channels; otherwise, one of the channels exhibits outage with probability one, and the system can perform no better than the case of no diversity with half the SNR. As a result, our analysis only considers the case $R_i \leq \ln(1+\text{SNR}/2)$. Moreover, due to the symmetry of the component channels, one can expect symmetric rates, *i.e.*, $R_1 = R_2 = R$, to be optimal; thus, we focus on this case. With these simplifications, we observe that, in contrast to the triangular outage regions for optimal parallel channel coding in Fig. 10-7, the rectangular outage regions for independent channel coding in Fig. 10-9 are well-matched to the on-off channel realizations.

Optimizing average distortion for the MD system requires a tradeoff between the distortion $D_1 = D_2$ achieved when only one description is received and the joint distortion $D_0$ achieved when both descriptions are received. Although this tradeoff is available in (10.30), we refactor it for our purposes here. Specifically, we set

$$D_1 = D_2 \approx \begin{cases} \exp(-(1-\lambda)2\beta R) \ , & 0 \leq \lambda < 1 \\ 1 \ , & \lambda = 1 \end{cases} , \tag{10.39}$$

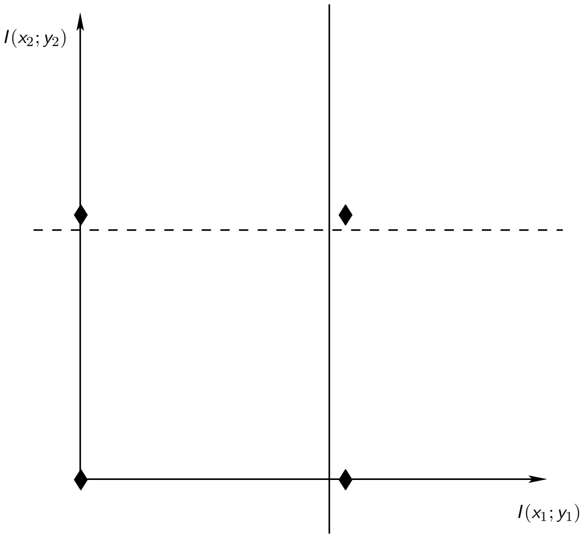


**Figure 10-9.** Outage region boundaries for MD source coding with independent channel coding. The ♦ symbols correspond to the sample mutual information pairs $(0,0)$, $(0,\ln(1+\text{SNR}/2))$, $(\ln(1+\text{SNR}/2),0)$, and $(\ln(1+\text{SNR}/2),\ln(1+\text{SNR}/2))$. The solid line corresponds to the outage region boundary for the first channel, and the dashed line corresponds to the outage region boundary for the second channel. The outage region for channel one (resp. channel two) is to the left (resp. below) the boundary.

where $R$ is the channel coding rate for a single channel. Thus, if $\lambda = 0$, the individual descriptions achieve the single description rate-distortion bound. With this parameterization of $D_1$ and $D_2$, the MD high-resolution approximation (10.30) yields

$$D_0 \approx \begin{cases} \frac{1}{2}\exp(-(1+\lambda)2\beta R) \ , & 0 \le \lambda < 1 \\ \exp(-4\beta R) \ , & \lambda = 1 \end{cases} \tag{10.40}$$

for the joint distortion when both descriptions are received. We note that an essentially identical approximation is developed in [76].

The minimum average distortion for source coding diversity is then approximately

$$\begin{aligned} D_{\text{SCDIV}} \approx \min\{ \min_{0<\lambda<1} \epsilon^2 + 2\epsilon(1-\epsilon)(1+\text{SNR}/2)^{-(1-\lambda)2\beta} + \frac{1}{2}(1-\epsilon)^2(1+\text{SNR}/2)^{-(1+\lambda)2\beta}, \\ [1-(1-\epsilon)^2] + (1-\epsilon)^2(1+\text{SNR}/2)^{-4\beta}\} \ . \end{aligned} \tag{10.41}$$

For $\lambda = 1$, source coding diversity performance reduces to that of channel coding diversity; for $\lambda = 0$, source coding diversity performance reduces to that of no diversity with half the SNR. Because optimization over $\lambda$ does not lend much insight, we delay discussion of source coding diversity quality- and outage-limited regimes to the next section, where we also compare with the other approaches.

## ■ 10.3.5 Comparison

Fig. 10-10 compares average distortion performance of source and channel coding diversity by displaying the minimum average distortions (10.33), (10.37), and (10.41) as functions of the component channel quality, SNR, in the on state, for different values of the probability of a component channel being off, $\epsilon$. The results in Fig. 10-10 are clearly consistent with our

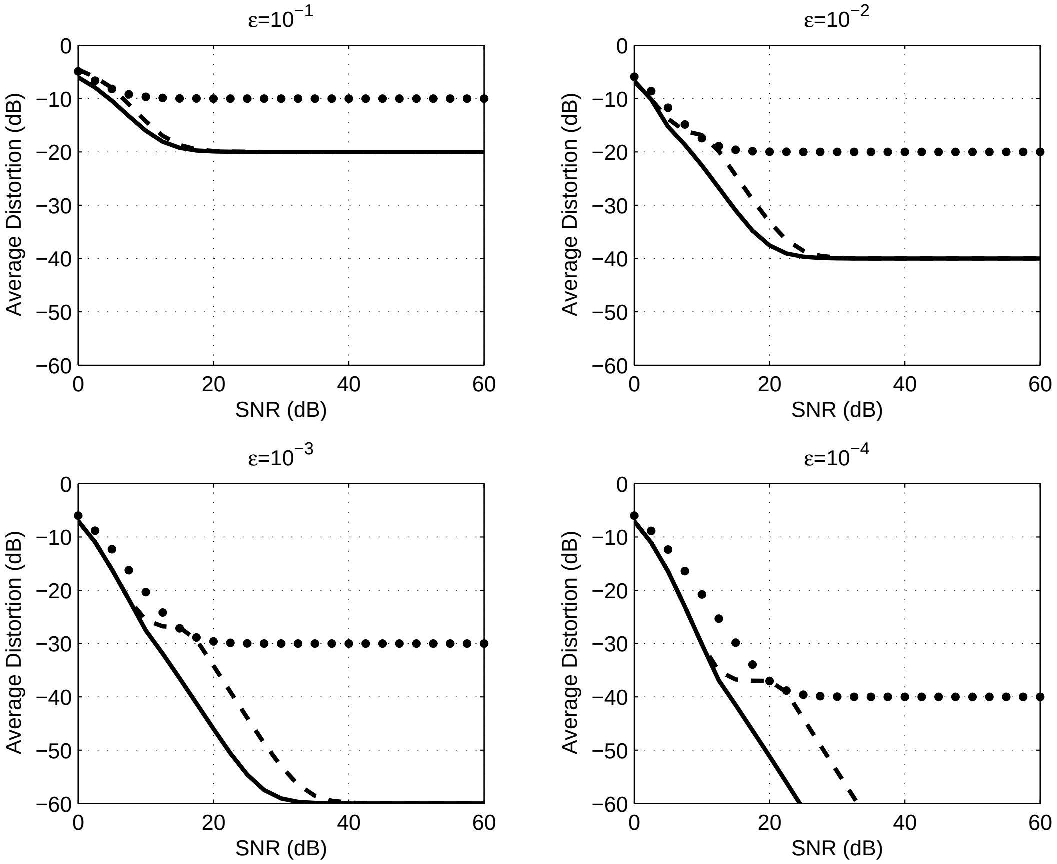


**Figure 10-10.** Average distortion performance over on-off channels. The plots show average distortion as a function of SNR; successively lower curves correspond to no diversity (dotted lines), optimal channel coding diversity (dashed lines), and source coding diversity (solid lines), respectively. Each plot corresponds to a different value for the probability $\epsilon$ of a component channel being off, and all are for $\beta = 1$.

intuitive discussion of source and channel coding diversity performance in Section 10.1.1. For moderate SNR, depending upon $\epsilon$, both systems exhibit transitions from $\mathrm{SNR}^{-4}$ behavior to $\mathrm{SNR}^{-2}$ behavior; however, the transition is generally less drastic for source coding diversity, especially for smaller $\epsilon$. The difference between the two systems is apparently the outage-limited behavior of the multiplexing mode for optimal channel coding diversity, for which the outage regions are not well-matched to the channel realizations. By contrast, the transition between the two quality-limited trends for source coding diversity is much less drastic, and this graceful degradation property of source coding diversity leads to their better performance over on-off channels. However, it is important to note that there is negligible difference between optimal channel coding diversity and source coding diversity at both low and high SNR.

## 10.4 Continuous State Channels

In cases where we are concerned with time or frequency selective multipath fading in a mobile radio channel or a range of possible interference levels in a cellular network, we can model the channel state $\boldsymbol{\theta}_i$ as taking on a continuum of values. For example, multiplicative fading is commonly modeled as a Rayleigh or Nakagami random variable in such scenarios. In the following section we study the average mean square distortion in the limit of high SNR for such continuous channels when the channel state is known to the receiver but not the transmitter. Since the distortion generally behaves as $\mathrm{SNR}^{-\Delta}$ for such channels, we are mainly interested in computing the distortion exponent defined as

$$\Delta = -\lim_{\mathrm{SNR}\to\infty} \frac{\log E[\mathsf{D}]}{\log \mathrm{SNR}}. \tag{10.42}$$

Note that there is an important difference between the average or transmit signal-to-noise ratio which is deterministic and known by both transmitter and receiver and the instantaneous or block signal-to-noise ratio which is random and known only at the receiver. Throughout the rest of the chapter, we always use SNR to refer to the former and consider the random, instantaneous signal-to-noise ratio as a random variable.

In Section 10.4.7, we plot the distortion exponents as well as the numerically computed average distortions for a Gaussian source transmitted over a complex Rayleigh fading additive white Gaussian noise channel. Hence the reader may find it useful to refer to Figures 10-11 and 10-12 as a concrete example for comparing the following results for the performance of each system.

### 10.4.1 Continuous Channel Model

For continuous state channels, the distribution of the mutual information random variable is generally difficult to compute exactly. For complex, additive white Gaussian noise channels with multiplicative fading, however, the mutual information random variable is $\mathsf{I} = \log(1 + \boldsymbol{\theta} \cdot \mathrm{SNR})$ where $\boldsymbol{\theta}$ corresponds to the multiplicative fading which is normalized so that $E[\boldsymbol{\theta}] = 1$ so that SNR is the transmit power or equivalently, the average received power. For $\boldsymbol{\theta} \cdot \mathrm{SNR} \gg 1$, we have

$$\mathsf{I} = \log(\boldsymbol{\theta} \cdot \mathrm{SNR}) + \log\left(1 + \frac{1}{\boldsymbol{\theta} \cdot \mathrm{SNR}}\right) = \log(\boldsymbol{\theta} \cdot \mathrm{SNR}) + O\left(\frac{1}{\boldsymbol{\theta} \cdot \mathrm{SNR}}\right) \approx \log(\boldsymbol{\theta} \cdot \mathrm{SNR})$$

and so $\exp \mathsf{I}$ is close to $\mathrm{SNR} \cdot \boldsymbol{\theta}$. [3] Thus, for additive Gaussian noise channels with multiplicative fading, we can develop asymptotic results by considering the first terms in the Taylor series expansion of the distribution of $\boldsymbol{\theta}$ near zero. More generally, we can focus on the high SNR limit by considering the Taylor series expansion of the distribution for the mutual information random variable for each channel.

Specifically, let $f_{\mathsf{I}}(t)$ and $F_{\mathsf{I}}(t)$ represent the probability density function (PDF) and cumulative distribution function (CDF) for the mutual information and let $f_{e^{\mathsf{I}}}(t)$ and $F_{e^{\mathsf{I}}}(t)$ represent the PDF and CDF for $\mathsf{I}$.[4] We consider the case where there exists a parameter called SNR such that

$$f_{e^{\mathsf{I}}}(t) \approx cp \left(\frac{t}{\mathrm{SNR}}\right)^{p-1} \quad \text{(with } p \geq 1\text{)} \tag{10.43}$$

and consequently $F_{e^{\mathsf{I}}}(t)$ can be approximated via

$$F_{e^{\mathsf{I}}}(t) \approx c \left(\frac{t}{\mathrm{SNR}}\right)^{p}. \tag{10.44}$$

Intuitively, SNR represents the transmit signal-to-noise ratio or the average signal-to-noise ratio and $F_{e^{\mathsf{I}}}(t)$ is the probability that the instantaneous signal-to-noise ratio is below $t$. As introduced in Section 10.2.5, the notion of approximation we use is that $a(\mathrm{SNR}) \approx b(\mathrm{SNR})$ if $\lim_{\mathrm{SNR}\to\infty} a(\mathrm{SNR})/b(\mathrm{SNR}) = 1$ and $\lim_{\mathrm{SNR}\to\infty} |a(\mathrm{SNR}) - b(\mathrm{SNR})| = 0$.

For example, in wireless communications, a common model is an additive white Gaussian noise channel with fading:

$$\mathsf{y}[i] = \boldsymbol{\theta} \cdot \mathsf{x}[i] + \mathsf{z}[i] \tag{10.45}$$

where $\boldsymbol{\theta}$ represents the fading and $\mathsf{z}[i]$ represents additive noise. A common approach is to obtain robustness by coding over two separate frequency bands or time-slots in which case the channel model becomes

$$\begin{aligned} \mathsf{y}_1[i] &= \boldsymbol{\theta}_1 \cdot \mathsf{x}_1[i] + \mathsf{z}_1[i] \\ \mathsf{y}_2[i] &= \boldsymbol{\theta}_2 \cdot \mathsf{x}_2[i] + \mathsf{z}_2[i]. \end{aligned}$$

If we are interested in Rayleigh fading then each $\boldsymbol{\theta}_i$ has an exponential distribution and at high SNR, the cumulative distribution function for $\exp \mathsf{I}(\mathbf{y}_i; \mathbf{x}_i)$ is approximated by $t/\mathrm{SNR}$ and hence the parameters $c$ and $p$ in (10.44) are both unity (*e.g.*, see [98,181] for a discussion of such high SNR expansions).

## 10.4.2 No Diversity

Perhaps the simplest case to consider is when there is only a single channel and no diversity is present. For such a scenario, a natural approach is cascading an SD source encoder/decoder $\mathrm{ENC}_{m\leftarrow\mathbf{s}}(\cdot)/\mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}}(\cdot)$ with a single channel encoder/decoder $\mathrm{ENC}_{\mathbf{x}\leftarrow m}(\cdot)/\mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\cdot)$. In terms of our general joint source-channel coding notation such a system has the encoder

[3] A similar expression can also be obtained for additive noise channels with non-Gaussian noise (*e.g.*, using techniques from [196,118]).

[4] Recall that we assume the mutual information optimizing input distribution is independent of the channel state. Hence it makes sense to speak of the mutual information distribution as given instead of a parameter controlled by the system designer.

and decoder

$$\mathbf{x} = \mathrm{ENC}_{\mathbf{x}\leftarrow\mathbf{s}}(\mathbf{s}) = \mathrm{ENC}_{\mathbf{x}\leftarrow m}(\mathrm{ENC}_{m\leftarrow\mathbf{s}}(\mathbf{s})) \tag{10.46a}$$

$$\hat{\mathbf{s}} = \mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\mathbf{y}}(\mathbf{y}) = \begin{cases} \mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}}(\mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\mathbf{y})), & \mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\mathbf{y}) \neq 0 \\ E[\mathbf{s}], & \text{otherwise.} \end{cases} \tag{10.46b}$$

**Theorem 34.** *The distortion exponent for a system with no diversity described by* (10.46) *is*

$$\Delta_{\mathrm{NO-DIV}} = \frac{2\beta p}{2\beta + p}\ , \tag{10.47}$$

*where $\beta$ is the processing gain defined in Section 10.2.4 and $p$ is the diversity order of the channel approximation in* (10.44).

*Proof.* The average distortion is

$$\begin{aligned} E[D] &= \min_D \Pr[I(x;y) < R(D)] + \{1 - \Pr[I(x;y) < R(D)]\} \cdot D && (10.48)\\ &= \min_D F_{e^{I}}(\exp R(D)) + [1 - F_{e^{I}}(R(D))] \cdot D && (10.49)\\ &\approx \min_D c\frac{D^{-p/(2\beta)}}{\mathrm{SNR}^p} + \left[1 - c\frac{D^{-p/(2\beta)}}{\mathrm{SNR}^p}\right] \cdot D && (10.50)\\ &\approx \min_D c\frac{D^{-p/(2\beta)}}{\mathrm{SNR}^p} + D. && (10.51) \end{aligned}$$

Differentiating and setting equal to 0 yields the minimizing distortion

$$D^* = \left(\frac{2\beta}{cp}\right)^{\frac{-2\beta}{2\beta+p}} \cdot \mathrm{SNR}^{\frac{-2\beta p}{2\beta+p}}.$$

Substituting this into (10.51) yields

$$E[D] \approx C_{\mathrm{NO-DIV}} \cdot \mathrm{SNR}^{\frac{-2\beta p}{2\beta+p}}. \tag{10.52}$$

where $C_{\mathrm{NO-DIV}}$ represents a term independent of SNR. Thus the distortion exponent is $2\beta p/(2\beta + p)$. □

## 10.4.3 Selection Channel Coding Diversity

Perhaps the simplest approach to using two independent channels is to use SD source coding with repetition channel coding and selection combining. In this scheme, the encoder quantizes the source, $\mathbf{s}$, to $\hat{\mathbf{s}}$, adds channel coding to produce $\mathbf{x}$, and repeats the result on both channels. The receiver decodes the higher quality channel and ignores the other. Formally, the encoder and decoder are given by

$$(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{ENC}_{\mathbf{x}_1,\mathbf{x}_2\leftarrow\mathbf{s}}(\mathbf{s}) = (\mathrm{ENC}_{\mathbf{x}\leftarrow m}(\mathrm{ENC}_{m\leftarrow\mathbf{s}}(\mathbf{s})), \mathrm{ENC}_{\mathbf{x}\leftarrow m}(\mathrm{ENC}_{m\leftarrow\mathbf{s}}(\mathbf{s}))) \tag{10.53a}$$

$$\hat{\mathbf{s}} = \mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\mathbf{y}_1,\mathbf{y}_2}(\mathbf{y}_1, \mathbf{y}_2) = \begin{cases} \mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}}(\mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\mathbf{y}_1)), & \mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\mathbf{y}_1) \neq 0 \\ \mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}}(\mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\mathbf{y}_2)), & \begin{array}{l}\mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\mathbf{y}_1) = 0 \\ \text{and } \mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\mathbf{y}_2) \neq 0\end{array} \\ E[\mathbf{s}], & \text{otherwise} \end{cases} \tag{10.53b}$$

where the SD source encoder/decoder and the single channel encoder/decoder are denoted

$$\mathrm{ENC}_{m\leftarrow\mathbf{s}}(\cdot)/\mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}}(\cdot) \text{ and } \mathrm{ENC}_{\mathbf{x}\leftarrow m}(\cdot)/\mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\cdot).$$

Thus, the quantized source signal will be recovered provided either channel is good. While such a scheme is sub-optimal in terms of resource use, it is simplest to understand and easiest to implement. The following theorem (proved in Appendix H.1) characterize asymptotic performance.

**Theorem 35.** *The distortion exponent for a system with selection channel coding diversity described by* (10.53) *is*

$$\Delta_{\mathrm{SEL-CCDIV}} = \frac{2\beta p}{\beta + p}. \tag{10.54}$$

### ■ 10.4.4 Multiplexed Channel Coding Diversity

A key drawback of repetition coding with selection combining is that it wastes the potential bandwidth of one channel in order to provide diversity. When the channel is usually good, such a scheme can be significantly sub-optimal. Hence, a complementary approach is channel multiplexing where the source is quantized using SD coding and this message is split over both channels. We define a channel multiplexing system as one with encoder and decoder given by

$$(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{ENC}_{\mathbf{x}_1,\mathbf{x}_2\leftarrow\mathbf{s}}(\mathbf{s}) = (\mathrm{ENC}_{\mathbf{x}_1\leftarrow m_1}(\mathrm{ENC}_{m_1\leftarrow\mathbf{s}}(\mathbf{s})), \mathrm{ENC}_{\mathbf{x}_2\leftarrow m_2}(\mathrm{ENC}_{m_2\leftarrow\mathbf{s}}(\mathbf{s}))) \tag{10.55a}$$

$$\hat{\mathbf{s}} = \mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\mathbf{y}_1,\mathbf{y}_2}(\mathbf{y}_1, \mathbf{y}_2) = \begin{cases} \mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}}(\mathrm{DEC}_{\hat{m}_1\leftarrow\mathbf{y}_1}(\mathbf{y}_1), \mathrm{DEC}_{\hat{m}_2\leftarrow\mathbf{y}_2}(\mathbf{y}_2)), & \mathrm{DEC}_{\hat{m}_1\leftarrow\mathbf{y}_1}(\mathbf{y}_1) \neq 0 \text{ and} \\ & \mathrm{DEC}_{\hat{m}_2\leftarrow\mathbf{y}_2}(\mathbf{y}_2) \neq 0 \\ E[\mathbf{s}], & \text{otherwise.} \end{cases} \tag{10.55b}$$

where the single channel encoders/decoders, the first and second half of the output of a single description source encoder, and the source decoder are denoted

$$\mathrm{ENC}_{\mathbf{x}_i\leftarrow m_i}(\cdot)/\mathrm{DEC}_{\hat{m}_i\leftarrow\mathbf{y}_i}(\cdot) \text{ and } \mathrm{ENC}_{m_i\leftarrow\mathbf{s}}(\cdot) \text{ and } \mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}}(\cdot).$$

If both channels are good enough to support successful decoding, then this scheme can transmit roughly twice the rate of a repetition coding system. The drawback is since either channel being bad can cause decoding failure, the system is less robust. The following theorem (proved in Appendix H.2) characterizes asymptotic performance.

**Theorem 36.** *The distortion exponent for a system with multiplexed channel coding diversity described by* (10.55) *is*

$$\Delta_{\mathrm{MPX-CCDIV}} = 4p\beta/(p + 4\beta). \tag{10.56}$$

Intuitively, we expect that when bandwidth is plentiful and outage is the dominating concern, the diversity provided by repetition coding is more important than the extra rate provided by channel multiplexing. When bandwidth is scarce, we expect the reverse to be true. We can verify this intuition by examining the distortion exponents in these two limits to obtain

$$\lim_{\beta/p\to\infty} \frac{\Delta_{\mathrm{SEL-CCDIV}}}{\Delta_{\mathrm{MPX-CCDIV}}} = 2 \tag{10.57}$$

$$\lim_{\beta/p\to 0} \frac{\Delta_{\mathrm{SEL-CCDIV}}}{\Delta_{\mathrm{MPX-CCDIV}}} = \frac{1}{2}. \tag{10.58}$$

The distortion exponents are equal if $p = 2\beta$.

## ■ 10.4.5 Optimal Channel Coding Diversity

Each of the previous schemes used SD source coding with some form of independent channel coding and hence was sub-optimal. With SD source coding, the optimal strategy is to use parallel channel coding. In this scheme, the two component channels are treated as a single parallel channel with channel encoding and decoding performed jointly over both. Specifically, we define optimal channel coding diversity as

$$(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{ENC}_{\mathbf{x}_1,\mathbf{x}_2\leftarrow\mathbf{s}}(\mathbf{s}) = \mathrm{ENC}_{\mathbf{x}\leftarrow m}(\mathrm{ENC}_{m\leftarrow\mathbf{s}}(\mathbf{s})) \tag{10.59a}$$

$$\hat{\mathbf{s}} = \mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\mathbf{y}_1,\mathbf{y}_2}(\mathbf{y}_1, \mathbf{y}_2) = \begin{cases} \mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}}(\mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\mathbf{y}_1, \mathbf{y}_2)), & \mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\mathbf{y}_1, \mathbf{y}_2) \neq 0 \\ E[\mathbf{s}], \text{ otherwise} \end{cases} \tag{10.59b}$$

where the SD source encoder/decoder and the parallel channel encoder/decoder are denoted

$$\mathrm{ENC}_{m\leftarrow\mathbf{s}}(\cdot)/\mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow\hat{m}}(\cdot) \text{ and } \mathrm{ENC}_{\mathbf{x}\leftarrow m}(\cdot)/\mathrm{DEC}_{\hat{m}\leftarrow\mathbf{y}}(\cdot).$$

Since parallel channel coding optimally uses the channel resources, it dominates both repetition coding with selection combining and channel multiplexing as characterized by the following theorem (proved in Appendix H.3).

**Theorem 37.** *The distortion exponent for a system with optimal channel coding diversity described by* (10.59) *is*

$$\Delta_{\mathrm{OPT-CCDIV}} = \frac{4p\beta}{p + 2\beta}. \tag{10.60}$$

## ■ 10.4.6 Source Coding Diversity

Next, we consider the case where the source is transmitted over a pair of independent channels using MD source coding. Specifically, we consider a system with

$$(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{ENC}_{\mathbf{x}_1,\mathbf{x}_2\leftarrow\mathbf{s}}(\mathbf{s}) = (\mathrm{ENC}_{\mathbf{x}_1\leftarrow m_1}(\mathrm{ENC}_{m_1\leftarrow\mathbf{s}}(\mathbf{s})), \mathrm{ENC}_{\mathbf{x}_2\leftarrow m_2}(\mathrm{ENC}_{m_2\leftarrow\mathbf{s}}(\mathbf{s}))) \tag{10.61a}$$

$$\hat{\mathbf{s}} = \mathrm{DEC}_{\hat{\mathbf{s}}\leftarrow \mathbf{y}_1,\mathbf{y}_2}(\mathbf{y}_1,\mathbf{y}_2) =$$

$$\begin{cases} \mathrm{DEC}_{\hat{\mathbf{s}}_1\leftarrow \hat{m}_1}(\mathrm{DEC}_{\hat{m}_1\leftarrow \mathbf{y}_1}(\mathbf{y}_1)), & \mathrm{DEC}_{\hat{m}_1\leftarrow \mathbf{y}_1}(\mathbf{y}_1) \neq 0 \text{ and} \\ & \mathrm{DEC}_{\hat{m}_2\leftarrow \mathbf{y}_2}(\mathbf{y}_2) = 0 \\ \mathrm{DEC}_{\hat{\mathbf{s}}_2\leftarrow \hat{m}_2}(\mathrm{DEC}_{\hat{m}_2\leftarrow \mathbf{y}_2}(\mathbf{y}_2)), & \mathrm{DEC}_{\hat{m}_1\leftarrow \mathbf{y}_1}(\mathbf{y}_1) = 0 \text{ and} \\ & \mathrm{DEC}_{\hat{m}_2\leftarrow \mathbf{y}_2}(\mathbf{y}_2) \neq 0 \\ \mathrm{DEC}_{\hat{\mathbf{s}}_{1,2}\leftarrow \hat{m}_{1,2}}(\mathrm{DEC}_{\hat{m}_1\leftarrow \mathbf{y}_1}(\mathbf{y}_1), \mathrm{DEC}_{\hat{m}_2\leftarrow \mathbf{y}_2}(\mathbf{y}_2)), & \mathrm{DEC}_{\hat{m}_1\leftarrow \mathbf{y}_1}(\mathbf{y}_1) \neq 0 \text{ and} \\ & \mathrm{DEC}_{\hat{m}_2\leftarrow \mathbf{y}_2}(\mathbf{y}_2) \neq 0 \\ E[\mathbf{s}], & \mathrm{DEC}_{\hat{m}_1\leftarrow \mathbf{y}_1}(\mathbf{y}_1) = 0 \text{ and} \\ & \mathrm{DEC}_{\hat{m}_2\leftarrow \mathbf{y}_2}(\mathbf{y}_2) = 0 \end{cases} \tag{10.61b}$$

where $\mathrm{ENC}_{m_1\leftarrow \mathbf{s}}(\cdot)$ and $\mathrm{ENC}_{m_2\leftarrow \mathbf{s}}(\cdot)$ represent the two quantizations of the source produced by the MD source coder, $\mathrm{DEC}_{\hat{\mathbf{s}}_i\leftarrow \hat{m}_i}(\cdot)$ represent the possible source decoders described in Table 10.1, and $\mathrm{ENC}_{\mathbf{x}_i\leftarrow m_i}(\cdot)$ / $\mathrm{DEC}_{\hat{m}_i\leftarrow \mathbf{y}_i}(\cdot)$ correspond to single channel encoders/decoders. The performance of such a system is characterized by Theorem 38 (proved in Appendix H.4).

**Theorem 38.** *The distortion exponent for source coding diversity as described by* (10.61) *is*

$$\Delta_{\mathrm{SCDIV}} = \max\left[\frac{8\beta p}{4\beta + 3p}, \frac{4\beta p}{4\beta + p}\right]. \tag{10.62}$$

When $p \leq 4\beta$, MD source coding achieves diversity in the sense that if either channel is bad but the other is good a coarse-grained description of the source can be reconstructed while if both channels are good, a fine-grained description can be reconstructed. Therefore, in this regime, source coding diversity dominates sub-optimal channel coding diversity because it takes advantage of the redundancy between descriptions at the source coding layer.

When $p \geq 4\beta$, however, the max in (10.62) selects the second term. In this regime, it is more important to maximizes the transmitted rate than protect against fading. Thus source coding diversity degenerates into multiplex channel coding diversity as analyzed in Section 10.4.4.

In both regimes, optimal channel coding diversity dominates source coding diversity.

## ■ 10.4.7 Rayleigh Fading AWGN Example

In this section, we evaluate the various distortion exponents on a complex Rayleigh fading additive white Gaussian noise (AWGN) channel. The high SNR approximation for the mutual information on each Rayleigh fading AWGN channel is $F_{e^I}(t) \approx (t/\mathrm{SNR})$, *i.e.*, $p = 1$ in (10.44) (*e.g.*, see [98,181] for a discussion of such high SNR expansions).

The resulting distortion exponents are summarized[5] in Table 10.2 and plotted in Fig. 10-11. When the processing gain is small (*i.e.*, $\beta \ll 1$), multiplex and optimal channel coding diversity as well as source coding diversity all approach a distortion exponent of $4\beta$, while selection channel coding diversity and no diversity both approach distortion exponents of $2\beta$. Intuitively, this occurs because since bandwidth is scarce, a good system should try to maximize the information communicated by sending different information on each channel.

[5] The distortion exponents in this chapter are slightly different than in [96] due to different definitions of the processing gain as described in Section 10.2.4.

Multiplex coding does this by sending different information on each channel using the same code, optimal channel coding does this by using a different code for each channel, and multiple descriptions coding does this by sending different source descriptions on each channel. Since neither selection diversity nor no diversity provide any multiplexing gain (in the sense of [204]) both of these systems achieve the same sub-optimal distortion exponent.

When the processing gain is large (*i.e.*, $\beta \gg 1$), selection and optimal channel coding diversity as well as source coding diversity all approach a distortion exponent of 2, while systems with multiplex channel coding diversity or no diversity achieve a smaller distortion exponent of 1. Intuitively, this occurs because, since bandwidth is plentiful, even one good channel provides plenty of rate to send a satisfactory description of the source. Thus good systems should try to maximize robustness by being able to decode even if one channel fails completely.

At both extremes of processing gain, the best distortion exponent can be achieved either by exploiting diversity at the physical layer via parallel channel coding or at the application layer via multiple description coding. In some sense, this suggests that both physical layer and application layer systems are flexible enough to incorporate the main principles of diversity for continuous channels. Other sub-optimal schemes such as selection channel coding diversity are less flexible in that they only incorporate a subset of the important principles of diversity and thus approach the best distortion exponent in at most one extreme of processing gain. For all processing gains, however, optimal channel coding diversity is superior to source coding diversity, suggesting that the application layer system is missing something. In Section 10.5, we show that the loss of source coding diversity is essentially caused by separating the process of channel decoding from source decoding.

**Table 10.2.** Distortion exponents.

| System | $\Delta$ |
|---|---|
| No Diversity (Section 10.4.2) | $2\beta/(2\beta+1)$ |
| Selection Channel Coding Diversity (Section 10.4.5) | $2\beta/(\beta+1)$ |
| Multiplex Channel Coding Diversity (Section 10.4.4) | $4\beta/(4\beta+1)$ |
| Optimal Channel Coding Diversity (Section 10.4.5) | $4\beta/(2\beta+1)$ |
| Source Coding Diversity (Section 10.4.6) | $\max[8\beta/(4\beta+3), 4\beta/(4\beta+1)]$ |

Fig. 10-12 shows the average distortion for various systems transmitting over complex Rayleigh fading AWGN channels with $\beta = 1$ where the parameters in the rate optimizations have been numerically computed for each system using the high SNR approximations. As the plot indicates, the difference in performance suggested by the asymptotic results in Table 10.2 becomes evident even at reasonable SNR. Indeed, as the figure shows, optimal channel coding diversity is always superior to source diversity and achieves an advantage of a few dB at moderate SNR. Source diversity is superior to selection diversity by a similar margin. In contrast, Fig. 10-10 shows that for on-off channels, source-diversity is always better than optimal channel coding diversity for on-off channels. Evidently, none of the systems considered so far are universally optimal and the best way to achieve diversity depends on the qualitative features of the channel. In the next section, we consider a joint source-channel coding system which we show achieves the benefits of source-diversity for on-off channels and the benefits of optimal channel diversity for continuous state channels.

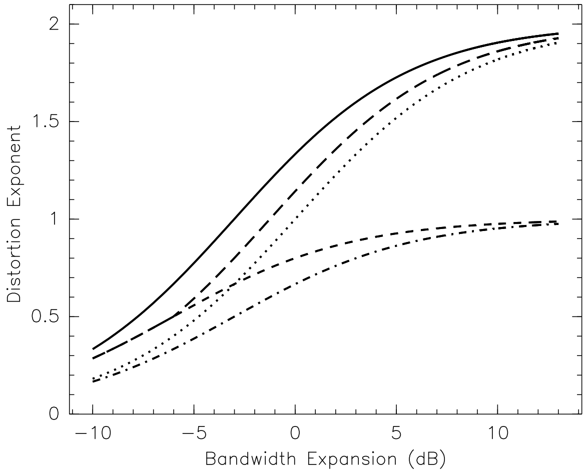


**Figure 10-11.** Distortion exponents as a function of bandwidth expansion factor $\beta$ in decibels. From top to bottom on the right hand side the curves correspond to optimal channel coding diversity (Section 10.4.5), source coding diversity (Section 10.4.6), selection channel coding diversity (Section 10.4.3), multiplexed channel coding diversity (Section 10.4.4), and no diversity (Section 10.4.2).

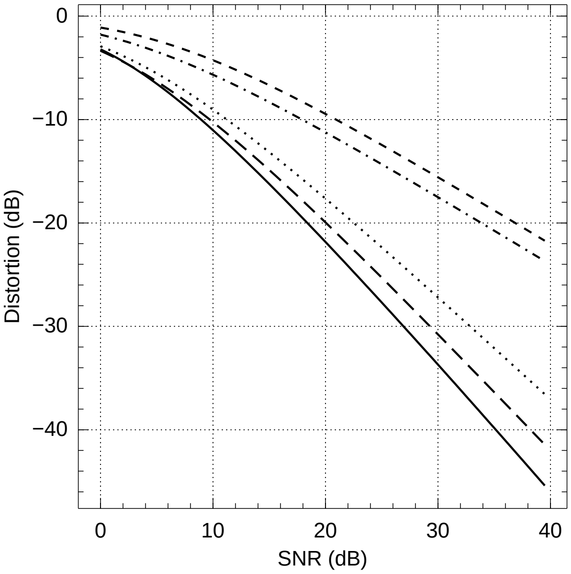


**Figure 10-12.** Average distortion performance on a complex Rayleigh fading additive white Gaussian noise channel with processing gain $\beta = 1$. From top to bottom on the right hand side the curves correspond to no diversity (Section 10.4.2), multiplexed channel coding diversity (Section 10.4.4), selection channel coding diversity (Section 10.4.3), source coding diversity (Section 10.4.6), and optimal channel coding diversity (Section 10.4.5).

## 10.5 Source Coding Diversity with Joint Decoding

In this section we consider source coding diversity with a joint decoder that uses the redundancy in both the source coder and channel coder to decode the received signal. Specifically, we define source coding diversity with joint decoding to have encoder and decoder

$$(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{ENC}_{\mathbf{x}_1,\mathbf{x}_2 \leftarrow \mathbf{s}}(\mathbf{s}) = (\mathrm{ENC}_{\mathbf{x}_1 \leftarrow m_1}(\mathrm{ENC}_{m_1 \leftarrow \mathbf{s}}(\mathbf{s})), \mathrm{ENC}_{\mathbf{x}_2 \leftarrow m_2}(\mathrm{ENC}_{m_2 \leftarrow \mathbf{s}}(\mathbf{s}))) \tag{10.63a}$$

$$\hat{\mathbf{s}} = \mathrm{DEC}_{\hat{\mathbf{s}} \leftarrow \mathbf{y}_1,\mathbf{y}_2}(\mathbf{y}_1, \mathbf{y}_2) \tag{10.63b}$$

where $\mathrm{ENC}_{\mathbf{x}_1 \leftarrow m_1}(\cdot)/\mathrm{ENC}_{\mathbf{x}_1 \leftarrow m_1}(\cdot)$ are single channel encoders (with potentially but not necessarily different codes), $\mathrm{ENC}_{m_1 \leftarrow \mathbf{s}}(\cdot)/\mathrm{ENC}_{m_2 \leftarrow \mathbf{s}}(\cdot)$ are MD source encoders, $\mathrm{DEC}_{\hat{\mathbf{s}} \leftarrow \mathbf{y}_1,\mathbf{y}_2}(\cdot)$ denotes a joint source-channel decoder to be described in the sequel.

The motivation for joint source-channel decoding is illustrated by considering the conceptual diagram of an MD quantizer in Fig. 10-13. Since the two quantization indexes $\mathrm{ENC}_{m_1 \leftarrow \mathbf{s}}(\mathbf{s})$ and $\mathrm{ENC}_{m_2 \leftarrow \mathbf{s}}(\mathbf{s})$ are correlated, the channel decoder should take this correlation into account. For example, if one channel is good and $\mathbf{y}_1$ is accurately decoded to $m_1 = \mathrm{ENC}_{m_1 \leftarrow \mathbf{s}}(\mathbf{s})$ this decreases the number of possible values for $m_2$ and makes decoding $\mathbf{y}_2$ easier.

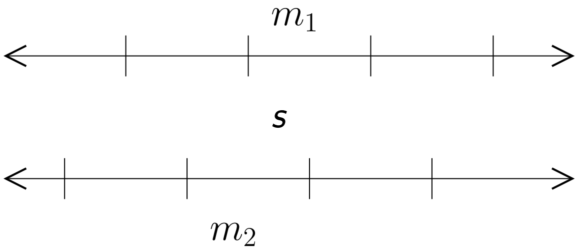


**Figure 10-13.** Conceptual diagram of an MD quantizer. The source $s$ is mapped to the quantizer bins labeled $m_1 = \mathrm{ENC}_{m_1 \leftarrow \mathbf{s}}(s)$ and $m_2 = \mathrm{ENC}_{m_2 \leftarrow \mathbf{s}}(s)$. Since only overlapping pairs of indexes are legal quantization values, if a receiver accurately decodes $m_1$ from the channel output $\mathbf{y}_1$, then there are only two possible values for $m_2$ in decoding a second channel output $\mathbf{y}_2$.

We show that a joint decoder that exploits this correlation can enlarge the region where both $m_1$ and $m_2$ are successfully decoded. Specifically, with separate decoding, both descriptions are decoded when both $I(\mathbf{x}_1; \mathbf{y}_1)$ and $I(\mathbf{x}_2; \mathbf{y}_2)$ exceed some rate threshold $R_T$, which is denoted as region III in Fig. 10-3. A joint decoder, however, also recovers both descriptions in region II yielding the decoding regions shown in Fig. 10-14. With these enlarged decoding regions, we show that source coding diversity with joint source-channel decoding achieves the same performance as optimal channel coding diversity for continuous channels in addition to providing the benefits of source coding diversity for on-off channels.

### 10.5.1 System Description

Next we describe one way to implement the architecture in (10.63) using an information theoretic formulation and random coding arguments.

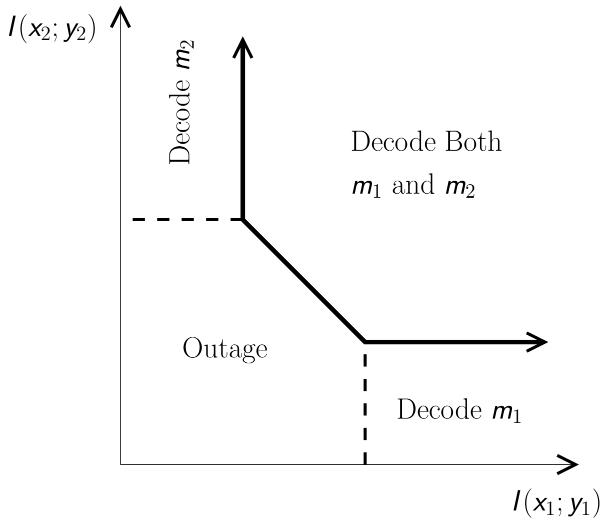


**Figure 10-14.** Decoding regions for a joint source-channel decoder.

### Source Encoding

Choose a test-channel distribution $p_{\hat{s}_1,\hat{s}_2|s}(\hat{s}_1, \hat{s}_2|s)$ with the marginal distributions

$$p_{\hat{s}_i}(\hat{s}_i) = \sum_{s,\hat{s}_{3-i}} p_{\hat{s}_1,\hat{s}_2|s}(\hat{s}_1, \hat{s}_2|s) p_s(s), \qquad \text{for } i \in \{1, 2\}. \tag{10.64}$$

Create a pair of rate $R$ random source codebooks, $\mathcal{C}_1$ and $\mathcal{C}_2$ by randomly generating $\exp n_s R$ sequences of length $n_s$ according to the i.i.d. test-channel distributions $p_{\hat{s}_i}(\hat{s}_i)$. To encode a source, find a pair of codewords $\hat{\mathbf{s}}_1 \in \mathcal{C}_1$, $\hat{\mathbf{s}}_2 \in \mathcal{C}_2$, such that the triple $(\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \mathbf{s})$ is strongly typical. According to [71], encoding will succeed with probability approaching one if[6]

$$R > I(\hat{s}_1; s) \tag{10.65a}$$

$$R > I(\hat{s}_2; s) \tag{10.65b}$$

$$2R > I(s; \hat{s}_1\hat{s}_2) + I(\hat{s}_1; \hat{s}_2). \tag{10.65c}$$

### Channel Encoding

For each channel, generate a rate $R$ random codebook, $\mathcal{C}_i$, by randomly selecting $\exp(n_s R)$ sequences (or equivalently $\exp(n_c R/\beta)$ sequences) of length $n_c$ according to the i.i.d. distribution $p_x(x)$. Encode the source codeword in the $i$th row of $\mathcal{C}_j$ by mapping it to the $i$th channel codeword in $\mathcal{C}_j$.

### Joint Decoding

Denote the output of channel $j$ as $\mathbf{y}_j$ for $j \in \{1, 2\}$. To decode, create the lists $\mathcal{L}_1$ and $\mathcal{L}_2$, by finding all channel codewords, $\mathbf{x}_j \in \mathcal{C}_j$, such that the pair $(\mathbf{x}_j, \mathbf{y}_j)$ is typical with respect to the distribution $p_{y_j,x_j|\boldsymbol{\theta}_j}(y, x|\theta_j)$. Next search for a unique pair of codewords $(\mathbf{x}_1, \mathbf{x}_2)$

[6] Note that [71] also includes a term $\hat{s}_0$ which can be ignored (*i.e.*, $\hat{s}_0$ can be set to null or set to a constant such as 0) for our purposes.

with $\mathbf{x}_1 \in \mathcal{C}_1$ and $\mathbf{x}_2 \in \mathcal{C}_2$ such that the corresponding source codewords $(\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2)$ are typical with respect to the distribution $p_{\hat{s}_1,\hat{s}_2}(\hat{s}_1, \hat{s}_2)$. If a unique pair is found, output the resulting source reconstructions. Otherwise declare a decoding error.

### Probability Of Error

The following theorem provides an achievable rate for source coding diversity with joint decoding.

**Theorem 39.** *Joint decoding will succeed with probability approaching one if*

$$\max\left[0, R - \beta \cdot I(x_1; y_1)\right] + \max\left[0, R - \beta \cdot I(x_2; y_2)\right] \leq I(\hat{s}_1; \hat{s}_2). \tag{10.66}$$

*Proof.* Decoding can fail if either the correct pair of source codewords are not typical or if an incorrect pair of source codewords are typical. According to the law of large numbers the probability of the former event tends to zero as the block length increases. Therefore, the union bound implies that if the probability of the latter tends to zero, then the total probability of a decoding error also tends to zero.

The probability that an incorrect pair of channel codewords is typical according to $p_{y_j,x_j|\boldsymbol{\theta}_j}(y, x|\theta_j)$ is roughly $\exp -n_c I(x_j; y_j)$. Since there are $\exp n_s R$ possible codewords for each channel, the expected list sizes are

$$|\mathcal{L}_j| = 1 + \exp\left[n_s R - n_c I(x_j; y_j)\right] + \epsilon \tag{10.67}$$

where the "1" corresponds to the correct channel codeword and $\epsilon$ denotes a quantity which goes to 0. Using standard arguments it is possible to show that the actual list sizes will be close to the expected list size with probability approaching one.

The probability that an incorrect pair of source codewords, $(\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2)$ corresponding to the channel codeword pair $(\mathbf{x}_1, \mathbf{x}_2)$ with $\mathbf{x}_j \in \mathcal{C}_j$ is typical is roughly $\exp -n_s I(\hat{s}_1; \hat{s}_2)$. Multiplying this probability by the number of incorrect pairs yields the expected number of incorrect codewords which are nonetheless typical:

$$\exp\left\{-n_s I(\hat{s}_1; \hat{s}_2) + \max\left[0, n_s R - n_c I(x_1; y_1)\right] + \max\left[0, n_s R - n_c I(x_2; y_2)\right]\right\}. \tag{10.68}$$

Therefore, after dividing through by $n_s$ and recalling that the processing gain is defined as $\beta = n_c/n_s$, we conclude that decoding succeeds provided that (10.66) holds. □

## ■ 10.5.2 Performance

In order to analyze performance, we must first choose a distribution for the source and channel codebooks. Naturally, we choose the capacity optimizing input distribution for each channel codebook $\mathcal{C}_j$. For the source codebook distribution we use a simpler form of the additive noise test-channel in [71]:

$$\hat{s}_j = s + n_j \tag{10.69}$$

where $(n_1, n_2)$ is a pair of zero-mean, variance $\sigma^2$, Gaussian random variables independent of $s$ and each other. For this distribution, the distortion when using only description $j$ is $D_j \leq \sigma^2$. When both descriptions are received they can be averaged to yield distortion $D_{1,2} \leq \sigma^2/2$.

### Performance on Continuous Channels

To derive the performance on continuous channels, we must choose $\sigma^2$ as a function of the channel parameters. The choice of $\sigma^2$ determines the rate and hence also the probability of outage and the distortion exponent. Our goal is to show that source coding diversity with joint decoding achieves the same distortion exponent as optimal channel coding diversity. Hence instead of solving an optimization problem to determine $\sigma^2$, we make an educated guess inspired by (H.30) to choose[7]

$$\sigma^2 = \mathrm{SNR}^{\frac{-4p\beta}{p+2\beta}}. \tag{10.70}$$

**Theorem 40.** *The distortion exponent for source coding diversity with joint decoding is at least as good as that for optimal channel coding diversity:*

$$\Delta_{\mathrm{SCDIV-JD}} \geq \Delta_{\mathrm{OPT-CCDIV}}. \tag{10.71}$$

Note that to achieve the distortion exponent in the previous theorem, the multiple description source redundancy is used in two qualitatively different ways. First, the redundnacy between $\mathbf{x}_1$ and $\mathbf{x}_2$ is used to recover the two source descriptions. In this sense, the source coding redundancy acts like channel coding redundancy in providing robustness to noise. Next, the redundancy between $\hat{\mathbf{s}}_1$ and $\hat{\mathbf{s}}_2$ is used to produce a better source reconstruction by averaging out the quantization noise. Thus, the benefit of joint source-channel decoding is that it can gain the maximum benefit of the redundancy required by multiple description coding both at the channel decoding stage and the source decoding stage.

## ■ 10.6 Concluding Remarks

We considered various architectures to minimize the average distortion in transmitting a source over independent parallel channels. Conceptually, we view the overall channel quality encountered by a system as a two-dimensional random variable where the two axes correspond to the Shannon mutual information for each channel. As illustrated in Fig. 10-3, the different architectures considered essentially correspond to systems which perform well when the channel quality is in a certain part of this two-dimensional mutual information plane. Thus minimizing the distortion for a given channel model corresponds to choosing an architecture matched to the shape of the overall channel mutual information distribution.

For on-off channel models, where a channel either fails completely or functions normally, the overall channel mutual information takes values on the Cartesian product of a finite set. This shape is well matched to source coding diversity, *i.e.*, MD source coding and independent channel coding, that exploits diversity at the application layer. Specifically, in the high SNR regime, it is essential that both channels carry redundant information so that if one channel fails the signal can still be decoded from the surviving channel. This forces channel coding diversity to use complete redundancy, and so the distortion when both channels are on is the same as when only one channel is on. In contrast, source coding diversity can use only partial redundancy by sending slightly different signals on each channel. When both channels are on, the differences in the two received descriptions lead to a higher resolution reconstruction and lower distortion. Therefore, source coding diversity

[7] Technically, it would be better to choose $\sigma^2$ to be proportional to the right hand side of (10.70) with a complicated proportionality constant. Since distortion exponent analysis essentially ignores constant factors, however, we ignore this refinement to simplify the exposition.

achieves substantially better performance than channel coding diversity as illustrated in Fig. 10-10.

In contrast, for fading, shadowing, and similar effects, the overall channel mutual information takes on a continuous range of values. This shape is better suited to optimal channel coding diversity that exploits at the physical layer. Specifically, in the high SNR regime, optimal channel coding diversity takes advantage of redundancy between the information transmitted across each channel while source coding diversity with separate decoding cannot. As one of our main results, we showed that for such channels the average distortion asymptotically behaves as $\mathrm{SNR}^{-\Delta}$. In particular, we calculated the distortion exponent $\Delta$ for various architectures and showed that the distortion exponent for optimal channel coding diversity is strictly better than for source coding diversity.

Finally, we demonstrated that there is no inherent flaw in source coding diversity on continuous channels. Instead, the inferior distortion exponent of source coding diversity is due to the sub-optimality of separate source and channel decoding. If joint source-channel decoding is allowed, source coding diversity achieves the same distortion exponent as optimal channel coding diversity. Thus, for the non-ergodic channels considered in this chapter, Shannon's source-channel separation theorem fails,[8] and the best overall performance is achieved by a joint source-channel architecture using multiple description coding.

Essentially, the architectural implications of our results can be summarized as follows. Application layer diversity is best for on-off channels, physical layer diversity is best for continuous-state channels, and (of the limited schemes we considered) joint source-channel coding achieves the best performance in either regime. Our results provide a way to balance the cost of choosing a potentially sub-optimal architecture against other concerns such as implementation complexity, ease of deployment, etc. Finally, our results on joint source-channel coding illustrate the price of layering incurred in separating source and channel coding.

While this chapter explores a variety of architectures, many aspects of the detailed design, analysis and implementation of such systems remain to be addressed. On the information theory side, determining the best possible average distortion, or at least lower bounds to the best distortion, would be a valuable step. Similarly, determining the performance for architectures using broadcast channel codes combined with successive refinement source codes, hybrid digital-analog codes, or other joint source-channel architectures would be interesting. Also, determining second-order performance metrics beyond the distortion exponent would be useful in designing practical systems. Some issues of interest in signal processing and communication theory include developing practical codes achieving the theoretical advantages of joint source-channel decoding, generalizing the results in this chapter to sources with memory or correlated channels (*e.g.*, as found in multiple antenna systems), and studying the effect of imperfect channel state information at the receiver. Finally, a wide array of similar questions arise in a variety of network problems such as relay channels, multi-hop channels, and interference channels. For network scenarios, both the number of possible architectures as well as the advantages of sophisticated systems will be larger.

[8] We believe that the main value of Shannon's original source-channel separation theorem was in showing that bits are a sufficient currency between source and channel coding systems. Thus even though the system in Section 10.5 has separate encoding and only the decoding is performed jointly, we say that the separation theorem breaks down because exchanging bits is no longer sufficient. Specifically, such a joint decoding system would need to pass lists, log-likelihood ratios, or similar information from the channel coding layer to the source coding layer.

Chapter 11

# Concluding Remarks

## ■ 11.1 Distortion Side Information Models

In the first part of this thesis we focused on dynamics in source coding. Essentially, our goal was to study scenarios where the relative importance or qualitative notion of distortion varies throughout the source signal. To do so, we introduced the idea of side information which affects the distortion measure but is independent of the source. We developed the theory of such distortion side information and showed that knowing it at the encoder is often as good as knowing it at both encoder and decoder. Furthermore, we constructed practical algorithms to exploit such side information using lattices and erasure quantization codes. If linear complexity erasure quantization codes (such as the ones in Chapter 4) are used, these constructions require only linear complexity to fully utilize the distortion side information.

Our results about distortion side information illustrate a new principle for use in lossy compression. Specifically, in addition to the boundary gain achieved by entropy coding and the granular gain achieved by choosing a good high-dimensional packing, distortion side information provides a way to exploit the "modelling gain" of a more accurate distortion measure. The fact that distortion side information based quantizers require only linear complexity suggests that this principle is one which can be practically implemented. Beyond the specific results presented in this thesis, we believe that exploring the role of dynamic distortion models may lead to a rich set of new possibilities for source coding.

First, while there is a growing understanding of perceptual effects in lossy compression, leveraging such knowledge can be difficult due to the complexity of incorporating it in classical quantizer design. In contrast, the framework presented in this thesis provides a clean interface between the tasks of constructing quantizers and investigating the properties of human perception. Biologists, psychologists, and cognitive scientists can debate and propose distortion side information models while engineers can focus on designing quantizers for generic distortion side information. By having a common interface, scientists can communicate physiologically meaningful models to engineers and engineers can suggest computationally attractive model structures for scientists to investigate.

Second, separating the quantizer structure from the detailed distortion model has a number of practical advantages in compression standards. To facilitate forward and backward compatibility, a common codebook format that supports a variety of distortion side information models could be standardized *without* fixing the exact distortion side information model. Then each source encoder could use a potentially different distortion side information model to produce a bit stream understandable to every decoder without the need to specify the model details. This would provide an incentive for quantizer designers to research better distortion models and reap the benefit via proprietary encoders. Alternatively, such a structure would allow constant innovations in the distortion model used at the encoder without the need to upgrade existing decoders.

Third, there is a growing interest in the interaction of compression with control and sensing [129, 30, 128, 170, 157, 132]. In such scenarios, it seems plausible that the benefit of incorporating the dynamics of the source in performing compression may outweigh the packing gain or boundary gain optimized by traditional quantizers. Specifically, the appropriate notion of distortion may depend on a controller's current state or on the noise or nonlinearities present in a sensor's observations. The corresponding quantizers may need to combine ideas from classical source coding theory and distortion side information with real-time structures common in control theory.

While we believe the results in this thesis represent a valuable step forward, many open issues remain in realizing the potential benefits of distortion side information. To crisply illustrate the advantages of the distortion side information framework we focused on uniform or high-resolution models where the source and side information are independent. In Sections 2.5.3 and 2.6 we explored the penalty for deviating from these assumptions and showed that distortion side information is still useful. But a better understanding of distortion side information in more general scenarios would be valuable. Specifically, investigating the design of optimal quantizers at finite-resolution (perhaps with a Lloyd-Max type algorithm), or for non-difference distortion measures would provide a clear practical benefit. Similarly, exploring the connection between estimation, distortion side information, and lossy compression may uncover interesting connections between information theory and control/sensing.

## ■ 11.2 Delay

Delay is an essential parameter in real-time systems and has received a great deal of attention in networking research. Much of this research focuses on the important problems of designing protocols, scheduling algorithms, and routing architectures to minimize latency. In contrast, relatively little attention has been devoted to understanding the fundamental limits of delay from a communication theory perspective or developing delay-optimal coding schemes.

This thesis presents several communication models focused on delay and examines the associated rate-delay-robustness trade-offs. The resulting analysis yields the fundamental limits of delay for various scenarios as well as explicit constructions for the corresponding delay-optimal codes. Such codes may be useful in a number of real-time applications, but they also illustrate two more general points. First, delay is a parameter like transmit power, rate, or distortion which can be analyzed and optimized using information theoretic tools. Second, the decoding delay is intimately connected to the code structure and designing codes to minimize delay is different from designing codes to maximize minimum distance, product distance, diversity exponent, etc. Indeed, one of the main contributions of this thesis is to establish that coding is useful in reducing delay and the appropriate codes must be tailored to the channel dynamics as opposed to classical block notions of code performance.

We believe these insights will be essential to designing coding schemes for real time systems for both the idealized models in this thesis as well as more detailed (and less analytically tractable) scenarios encountered in practice. In particular, one can view the burst correcting theory/codes from Chapters 7 and 8 as one extreme and the theory/codes from Chapter 9 as another. In practice, we expect that appropriate codes would be constructed by combining these structures with each other as well as with other structures yet to be discovered.

For example, one possibility for future work would be to construct a burst correcting code which meets the optimal burst-delay trade-off in Chapter 7 but also corrects other non-bursty dynamics with a slightly longer delay. Another possibility would be to consider a multi-burst adversarial channel where $N$ bursts of length $\mathtt{B}$ could occur and analyze the decoding delay required as a function of the rate. A third possibility would be to require a code to meet the optimal burst-delay trade-off for bursts of length $\mathtt{B}$ and investigate the delay required if instead a burst of length $\mathtt{B}' > \mathtt{B}$ occurs. We believe that studying such generalizations could yield further interesting random coding structures like the one in Chapter 7 as well as practical codes such as those in Chapter 8.

At a higher level, we believe that many problem models and heuristic algorithms are essentially motivated by delay and formalizing the role of delay may yield interesting results. For example, multiple description coding is often motivated by citing delay constraints [11, 75,4,133]. It would be interesting to reformulate the classic multiple descriptions problem as a joint source-channel coding problem with delay constraints and investigate the resulting solution. We expect that optimal codes may contain aspects of multiple description coding combined with features of the codes in Chapters 7–9 and in [147, 148, 152].

In some sense, this is the motivation for exploring various diversity architectures in Chapter 10. In that setting, we consider a delay constraint of zero requiring the source data transmitted over a parallel channel at time $i$ to be decoded at time $i$. The main issue is whether to exploit the inherent diversity of the parallel channel at the physical layer through channel coding or at the application layer through source coding. We show that the former is better for continuous state channels, while the latter is better for on-off channels, and a joint source-channel decoding architecture achieves the best qualities of both.

Thus one advantage of explicitly including delay in problem formulations is that doing so illustrates the potential advantages of joint source-channel coding in a quantitative manner. Specifically, when delay is ignored, Shannon's source-channel separation theorem states that there is no fundamental loss in optimality by layering separately designed source and channel coders. In contrast, considering delay illustrates the price of layering by illustrating that a layered system based on existing source and channel coding ideas performs measurably worse for either on-off channels or continuous-state channels.

A natural extension to the work in Chapter 10 would be to allow coding over multiple blocks in time subject to some delay constraint as in Chapter 7 and investigate the performance of various architectures. Another extension would be to consider source and channel coding in concert with additional network layers and investigate the performance of the resulting architectures. Using the simplified asymptotic analysis of the distortion exponent framework may help bridge the gap between networking and information theory discussed in [56].

Interesting delay-motivated problem formulations are also possible in pure source coding settings. For example, consider a pair of distributed sensors whose observations are usually correlated so that multi-terminal source coding techniques such as Wyner-Ziv coding or Slepian-Wolf coding [46] can be used to reduce the required bit rate. If occasionally the correlation weakens or disappears for short bursts of time, what are the fundamental limits of the decoding delay and what types of codes achieve the optimal delay? We conjecture that it is possible to formulate and solve such problems using techniques analogous to those in Chapters 7–9.

Appendix A

# Notation Summary

The following is a brief summary of the notation used:

**Sequences:** Sequences are denoted as a variable indexed using brackets: $x\,[i]$, $y\,[i]$.

**Sub-sequences:** A sub-sequence indexed from $i$ to $j$ is denoted by stacking: $x\,[{}^{j}_{i}] = x\,[i]\,, x\,[i+1]\,, \ldots, x\,[j]$.

**Vectors and Matrices:** Vectors are denoted using bold face: $\mathbf{x}$, $\mathbf{y}$. When a sequence is referred to in its entirity it is denoted in bold, but elements of the sequence are only denoted as bold if the sequence element is a vector Thus, $\mathbf{x}$ denotes a sequence, $x\,[i]$ denotes a scalar element of a sequence, and $\mathbf{x}\,[i]$ denotes a vector element of a sequence. Matrices are denoted in uppercase bold: $\mathbf{C}$.

**Random Variables:** Random variables are denoted in the sans serif font: $\mathsf{x}$, $\mathsf{x}\,[i]$, $\mathsf{y}\,[i]$

**Random Vectors and Matrices:** Random vectors are denoted in bold sans serif font: $\boldsymbol{\mathsf{x}}\,[i]$, $\boldsymbol{\mathsf{y}}\,[i]$. Random matrices are denoted in uppercase bold sans serif: $\boldsymbol{\mathsf{C}}$.

**Vector Components:** Vector components are generally denoted using subscripts: $x_j\,[i]$, $\mathsf{x}_j\,[i]$.

**Parameters:** Parameters or resource constraints a system must satisfy are denoted with the typewriter font: $\mathtt{T}$, $\mathtt{B}$.

**Sets:** Calligraphic letters denote sets (*e.g.*, $s \in \mathcal{S}$) with $|\cdot|$ denoting the cardinality of its argument.

**Information and Entropy:** We denote mutual information and differential entropy as $I(\mathsf{x};\mathsf{y})$ and $h(\mathsf{x})$. A related quantity called information debt and denoted $\mathsf{I}_d[i,\mathtt{R}|\boldsymbol{\theta}]$ is defined in (9.9) of Section 9.3.1.

**Type Classes:** For any length $n$ sequence $q\,[{}^{n}_{1}]$, we denote the number of occurrences of the symbol $i$ as $\#[q\,[{}^{n}_{1}] \sim i]$. In the Information Theory literature, $\#[q\,[{}^{n}_{1}] \sim i]$ is sometimes refereed to as the type of $q\,[{}^{n}_{1}]$.

Appendix B

# Distortion Side Information Proofs

## ■ B.1 Group Difference Distortion Measures Proof

*Proof of Theorem 1:* Assume that $p^*_{\hat{\mathbf{s}}|\mathbf{s},\mathbf{q}}(\hat{s}|s,q)$ is an optimal test-channel distribution with the conditional $p^*_{\hat{\mathbf{s}}|\mathbf{q}}(\hat{s}|q)$. By symmetry, for any $t \in \mathcal{S}$, the shifted distribution

$$p^t_{\hat{\mathbf{s}}|\mathbf{s},\mathbf{q}}(\hat{s}|s,q) \stackrel{\Delta}{=} p^*_{\hat{\mathbf{s}}|\mathbf{s},\mathbf{q}}(\hat{s} \oplus t|s \oplus t, q) \tag{B.1}$$

must also be an optimal test-channel. Since mutual information is convex in the test-channel distribution, we obtain an optimal test-channel distribution, $p^{**}$, by averaging $t$ over $\mathcal{S}$ via the uniform measure $d_{\mathcal{S}}(t)$:

$$p^{**}_{\hat{\mathbf{s}}|\mathbf{s},\mathbf{q}}(\hat{s}|s,q) \stackrel{\Delta}{=} \int_{\mathcal{S}} p^t_{\hat{\mathbf{s}}|\mathbf{s},\mathbf{q}}(\hat{s}|s,q) d_{\mathcal{S}}(t). \tag{B.2}$$

To prove that the resulting distribution for $\hat{\mathbf{s}}$ given $\mathbf{q}$ is independent of $\mathbf{q}$, we will show that $p^{**}_{\hat{\mathbf{s}}|\mathbf{q}}(\hat{s}|q) = p^{**}_{\hat{\mathbf{s}}|\mathbf{q}}(\hat{s} \oplus r|q)$ for any $r \in \mathcal{S}$:

$$\begin{aligned}
p^{**}_{\hat{\mathbf{s}}|\mathbf{q}}(\hat{s}|q) &= \int_{\mathcal{S}} p^{**}_{\hat{\mathbf{s}}|\mathbf{s},\mathbf{q}}(\hat{s}|s,q) d_{\mathcal{S}}(s) && \text{(B.3)}\\
&= \int_{\mathcal{S}}\int_{\mathcal{S}} p^t_{\hat{\mathbf{s}}|\mathbf{s},\mathbf{q}}(\hat{s}|s,q) d_{\mathcal{S}}(t) d_{\mathcal{S}}(s) && \text{(B.4)}\\
&= \int_{\mathcal{S}}\int_{\mathcal{S}} p^*_{\hat{\mathbf{s}}|\mathbf{s},\mathbf{q}}(\hat{s} \oplus t|s \oplus t, q) d_{\mathcal{S}}(t) d_{\mathcal{S}}(s) && \text{(B.5)}\\
&= \int_{\mathcal{S}}\int_{\mathcal{S}} p^*_{\hat{\mathbf{s}}|\mathbf{s},\mathbf{q}}(\hat{s} \oplus r \oplus t|s \oplus r \oplus t, q) d_{\mathcal{S}}(r \oplus t) d_{\mathcal{S}}(s) && \text{(B.6)}\\
&= \int_{\mathcal{S}}\int_{\mathcal{S}} p^*_{\hat{\mathbf{s}}|\mathbf{s},\mathbf{q}}(\hat{s} \oplus r \oplus t|s \oplus r \oplus t, q) d_{\mathcal{S}}(t) d_{\mathcal{S}}(s \oplus r) && \text{(B.7)}\\
&= \int_{\mathcal{S}}\int_{\mathcal{S}} p^*_{\hat{\mathbf{s}}|\mathbf{s},\mathbf{q}}(\hat{s} \oplus r \oplus t|s \oplus t, q) d_{\mathcal{S}}(t) d_{\mathcal{S}}(s) && \text{(B.8)}\\
&= p^{**}_{\hat{\mathbf{s}}|\mathbf{q}}(\hat{s} \oplus r|q). && \text{(B.9)}
\end{aligned}$$

Equation (B.3) follows from Bayes' law and the fact that $p_{\mathbf{s}|\mathbf{q}}$ is the uniform measure on $\mathcal{S}$. The next two lines follow from the definition of $p^**$ and $p^t$ respectively. To obtain (B.6), we make the change of variable $t \to r \oplus t$, and then apply the fact that the uniform measure is shift invariant to obtain (B.7). Similarly, we make the change of variable $s \oplus r \to s$ to obtain (B.8).

Note that this argument applies regardless of whether the side information is available at the encoder, decoder, both, or neither. □

## ■ B.2 Binary-Hamming Rate-Distortion Derivations

In this section we derive the rate-distortion functions for a binary source with Hamming distortion.

### ■ B.2.1 With Encoder Side Information

The rate-distortion function for source independent side information available at the encoder only is the same as with the side information available at both encoder and decoder. Hence, we compute $R[\text{Q-ENC-W-NULL}](D)$ and $R[\text{Q-BOTH-W-NULL}](D)$ by considering the latter case and noting that optimal encoding corresponds to simultaneous description of independent random variables [46, Section 13.3.3]. Specifically, the source samples for each value of $\boldsymbol{q}$ can be quantized separately using the distribution

$$p_{\hat{s}|s,\boldsymbol{q}}(\hat{s}|s,q) = \begin{cases} 1-p_q, & \hat{s}=s \\ p_q, & \hat{s}=1-s. \end{cases} \tag{B.10}$$

The cross-over probabilities, $p_q$, correspond to the bit allocations for each value of the side information and are obtained by solving a constrained optimization problem:

$$R[\text{Q-BOTH-W-NULL}](D) = \min_{E[d(\boldsymbol{s},\hat{\boldsymbol{s}};\boldsymbol{q})=D]} \sum_{i=1}^{N} E[1-H_b(p_{\boldsymbol{q}})] \tag{B.11}$$

where $H_b(\cdot)$ is the binary entropy function.

Using Lagrange multipliers, we construct the functional

$$J(D) = \sum_{i=1}^{N} p_{\boldsymbol{q}}(i)\cdot[1+p_i\log p_i + (1-p_i)\log(1-p_i)] + \lambda\sum_{i=1}^{N} p_{\boldsymbol{q}}(i)\cdot[\alpha_i + p_i\beta_i].$$

Differentiating with respect to $p_i$ and setting equal to 0, yields

$$\frac{\delta J}{\delta p_i} = p_{\boldsymbol{q}}(i)\log\frac{p_i}{1-p_i} + \lambda p_{\boldsymbol{q}}(i)\beta_i = 0 \tag{B.12}$$

$$\log\frac{p_i}{1-p_i} = -\lambda s_i \tag{B.13}$$

$$p_i = \frac{2^{-\lambda\beta_i}}{1+2^{-\lambda\beta_i}}. \tag{B.14}$$

Thus we obtain the rate-distortion functions

$$R[\text{Q-ENC-W-NULL}](D) = R[\text{Q-BOTH-W-NULL}](D) = 1 - \sum_{i=1}^{N} p_{\boldsymbol{q}}(i)\cdot H_b\left(\frac{2^{-\lambda\beta_i}}{1+2^{-\lambda\beta_i}}\right) \tag{B.15a}$$

where $\lambda$ is chosen to satisfy

$$\sum_{i=1}^{N} p_{\boldsymbol{q}}(i)\left[\alpha_i + \beta_i\cdot\frac{2^{-\lambda\beta_i}}{1+2^{-\lambda\beta_i}}\right] = D. \tag{B.15b}$$

## ■ B.2.2 Without Encoder Side Information

When no encoder side information is available, decoder side information is useless. Hence, the problem is equivalent to quantizing a symmetric binary source with distortion measure

$$d(s,\hat{s}) = E[\alpha_{\boldsymbol{q}} + \beta_{\boldsymbol{q}}\cdot d_H(s,\hat{s})] = E[\alpha_{\boldsymbol{q}}] + E[\beta_{\boldsymbol{q}}]\cdot d_H(s,\hat{s}). \tag{B.16}$$

The rate-distortion function is obtained by scaling and translating the rate-distortion function for the classical binary-Hamming case:

$$R[\text{Q-NONE-W-NULL}](D) = 1 - H_b\left(\frac{D-E[\alpha_{\boldsymbol{q}}]}{E[\beta_{\boldsymbol{q}}]}\right) \tag{B.17}$$

## ■ B.3 High-Resolution Proofs

*Proof of Theorem 3:* To obtain $R[\text{Q-ENC-W-DEC}](D)$ we apply the Wyner-Ziv rate-distortion formula[1] to the "super-source" $\mathbf{s}' = (\mathbf{s}, \mathbf{q})$ yielding

$$R[\text{Q-ENC-W-DEC}](D) = \inf_{p_{\hat{\mathsf{s}}|\mathsf{s},\mathsf{q}}(\hat{s}|s,q)} I(\hat{\mathsf{s}}, \mathsf{q}; \mathsf{s}|\mathsf{w}) \tag{B.18}$$

where the optimization is subject to the constraint that $E[d(\mathsf{s}, v(\hat{\mathsf{s}}, \mathsf{w}); \mathsf{q})] \leq D$ for some reconstruction function $v(\cdot,\cdot)$. To obtain $R[\text{Q-BOTH-W-BOTH}](D)$ we specialize the well-known conditional rate-distortion function to our notation yielding

$$R[\text{Q-BOTH-W-BOTH}](D) = \inf_{p_{\hat{\mathsf{s}}|\mathsf{s},\mathsf{q},\mathsf{w}}(\hat{s}|s,q,w)} I(\hat{\mathsf{s}}; \mathsf{s}|\mathsf{w}, \mathsf{q}) \tag{B.19}$$

where the optimization is subject to the constraint that $E[d(\mathsf{s}, \hat{\mathsf{s}}; \mathsf{q})] \leq D$.

Let us define $\hat{\mathsf{s}}^*$ as the distribution which optimizes (B.18). Similarly, define $\hat{\mathsf{s}}^*_{\mathsf{w}}$ as the distribution which optimizes (B.19). Finally, define $\mathsf{z}$ given $\mathsf{q} = q$ to be a random variable with a conditional distribution that maximizes $h(\mathsf{z}|\mathsf{q} = q)$ subject to the constraint that

$$E[d(\mathsf{s}, \mathsf{s} + \mathsf{z}; \mathsf{q})|\mathsf{q} = q] \leq E[d(\mathsf{s}, \hat{\mathsf{s}}^*_{\mathsf{w}}; \mathsf{q})|\mathsf{q} = q]. \tag{B.20}$$

Then we have the following chain of inequalities:

$$\begin{aligned}
\Delta R(D) &\overset{\Delta}{=} R[\text{Q-ENC-W-DEC}](D) - R[\text{Q-BOTH-W-BOTH}](D) && \text{(B.21)}\\
&= I(\hat{\mathsf{s}}^*; \mathsf{q}, \mathsf{s}|\mathsf{w}) - [h(\mathsf{s}|\mathsf{q}, \mathsf{w}) - h(\mathsf{s}|\mathsf{q}, \mathsf{w}, \hat{\mathsf{s}}^*_{\mathsf{w}})] && \text{(B.22)}\\
&= I(\hat{\mathsf{s}}^*; \mathsf{q}, \mathsf{s}|\mathsf{w}) - h(\mathsf{s}|\mathsf{q}, \mathsf{w}) + h(\mathsf{s} - \hat{\mathsf{s}}^*_{\mathsf{w}}|\mathsf{q}, \mathsf{w}, \hat{\mathsf{s}}^*_{\mathsf{w}}) && \text{(B.23)}\\
&\leq I(\hat{\mathsf{s}}^*; \mathsf{q}, \mathsf{s}|\mathsf{w}) - h(\mathsf{s}|\mathsf{q}, \mathsf{w}) + h(\mathsf{s} - \hat{\mathsf{s}}^*_{\mathsf{w}}|\mathsf{q}) && \text{(B.24)}\\
&\leq I(\hat{\mathsf{s}}^*; \mathsf{q}, \mathsf{s}|\mathsf{w}) - h(\mathsf{s}|\mathsf{q}, \mathsf{w}) + h(\mathsf{z}|\mathsf{q}) && \text{(B.25)}\\
&\leq I(\mathsf{s} + \mathsf{z}; \mathsf{q}, \mathsf{s}|\mathsf{w}) - h(\mathsf{s}|\mathsf{q}, \mathsf{w}) + h(\mathsf{z}|\mathsf{q}) && \text{(B.26)}\\
&= h(\mathsf{s} + \mathsf{z}|\mathsf{w}) - h(\mathsf{s} + \mathsf{z}|\mathsf{w}, \mathsf{q}, \mathsf{s}) - h(\mathsf{s}|\mathsf{q}, \mathsf{w}) + h(\mathsf{z}|\mathsf{q}) && \text{(B.27)}\\
&= h(\mathsf{s} + \mathsf{z}|\mathsf{w}) - h(\mathsf{s}|\mathsf{q}, \mathsf{w}) && \text{(B.28)}\\
&= h(\mathsf{s} + \mathsf{z}|\mathsf{w}) - h(\mathsf{s}|\mathsf{w}) && \text{(B.29)}\\
\lim_{D \to D_{\min}} \Delta R(D) &= 0. && \text{(B.30)}
\end{aligned}$$

Equation (B.25) follows from the definition of $\mathsf{z}$ to be entropy maximizing subject to a distortion constraint. Since $\mathsf{z}$ is independent of $\mathsf{s}$ and $\mathsf{w}$, the choice $\hat{\mathsf{s}} = \mathsf{s} + \mathsf{z}$ with $v(\hat{s}, w) = \hat{s}$ is an upper bound to (B.18) and yields (B.26). We obtain (B.29) by recalling that according to our problem model in (2.2), $\mathsf{q}$ and $\mathsf{s}$ are independent given $\mathsf{w}$. Finally, we obtain (B.30) by using the "continuity of entropy" result from [109, Theorem 1].

Note that although the $\mathsf{z}$ in [109, Theorem 1] is an entropy maximizing distribution while our $\mathsf{z}$ is a mixture of entropy maximizing distributions, the special form of the density is not required for the continuity of entropy result in [109, Theorem 1]. To illustrate this, we show how to establish the continuity of entropy directly for any distortion measure where $D \to D_{\min} \Rightarrow \text{Var}[\mathsf{z}] \to 0$. One example of such a distortion measure is obtained if we choose $d(s, \hat{s}; q) = q \cdot |s - \hat{s}|^r$ with $r > 0$ and $\Pr[\mathsf{q} = 0] = 0$. Denoting $\text{Var}[\mathsf{z}|\mathsf{w}]$ as $\sigma^2_{\mathsf{z}|\mathsf{w}}$

[1] Some readers may be more familiar with the Wyner-Ziv formula as a difference of mutual informations (*e.g.*, as in [44]), but the form in (B.18) is equally valid [190] is sometimes more convenient.

and $\mathrm{Var}\left[s|w\right]$ as $\sigma^2_{s|w}$ and letting $\mathcal{N}(\alpha)$ represent Gaussian random variable with variance $\alpha$ yields

$$\limsup_{D\to D_{\min}} h(s+z|w)-h(s|w) = \limsup_{\sigma^2\to 0} h(s+z|w)-h(s|w) \tag{B.31}$$

$$\begin{aligned}
&= \limsup_{\sigma^2\to 0} h(s+z|w) \pm h(\mathcal{N}(\sigma^2_{s|w}+\sigma^2_{z|w})|w) \\
&\quad \pm h(\mathcal{N}(\sigma^2_{s|w})|w) - h(s|w) && \text{(B.32)}\\
&= \limsup_{\sigma^2\to 0} D(s|w||\mathcal{N}(\sigma^2_{s|w})) - D(s+z|w||\mathcal{N}(\sigma^2_{s|w}+\sigma^2_{z|w})) \\
&\quad + h(\mathcal{N}(\sigma^2_{s|w}+\sigma^2_{z|w})|w) - h(\mathcal{N}(\sigma^2_{s|w})|w) && \text{(B.33)}\\
&\leq D(s|w||\mathcal{N}(\sigma^2_{s|w})) - D(s|w||\mathcal{N}(\sigma^2_{s|w})) \\
&\quad + \limsup_{\sigma^2\to 0} [h(\mathcal{N}(\sigma^2_{s|w}+\sigma^2_{z|w})|w) - h(\mathcal{N}(\sigma^2_{s|w})|w)] && \text{(B.34)}\\
&= \limsup_{\sigma^2\to 0} \int \Big[h(\mathcal{N}(\sigma^2_{s|w}+\sigma^2_{z|w})|w=w) - h(\mathcal{N}(\sigma^2_{s|w})|w=w)\Big]\, dp_w(w) && \text{(B.35)}\\
&= \int \Big[\limsup_{\sigma^2\to 0} h(\mathcal{N}(\sigma^2_{s|w}+\sigma^2_{z|w})|w=w) - h(\mathcal{N}(\sigma^2_{s|w})|w=w)\Big]\, dp_w(w) && \text{(B.36)}\\
&= 0. && \text{(B.37)}
\end{aligned}$$

We obtain (B.33) since for any random variable $v$, the relative entropy from $v$ to a Gaussian takes the special form $D(v||\mathcal{N}(\mathrm{Var}\left[v\right])) = h(\mathcal{N}(\mathrm{Var}\left[v\right])) - h(v)$ [46, Theorem 9.6.5]. To get (B.34) we use the fact that relative-entropy (and also conditional relative-entropy) is lower semi-continuous [49]. This could also be shown by applying Fatou's Lemma [1, p.78] to get that if the sequences $p_1(x), p_2(x), \ldots$ and $q_1(x), q_2(x), \ldots$ converge to $p(x)$ and $q(x)$ then

$$\liminf \int p_i(x)\log[p_i(x)/q_i(x)] \geq \int p(x)\log[p(x)/q(x)].$$

Switching the lim sup and integral in (B.36) is justified by Lebesgue's Dominated Convergence Theorem [1, p.78] since the integrand is bounded for all values of $w$. In general, this bound is obtained from combining the technical condition requiring $h(s|w=w)$ to be finite with the entropy maximizing distribution in (2.20) and the expected distortion constraint in (2.21) to bound $h(s+z|q=q)$. For scaled quadratic distortions, $h(s+z|q=q)$ can be bounded above by the entropy of a Gaussian with the appropriate variance. To obtain (B.37) we first note that $\mathrm{Var}\left[z\right]\to 0$ implies $\mathrm{Var}\left[z|w=w\right]\to 0$ except possibly for a set of $w$ having measure zero. This set of measure zero can be ignored because the integrand is finite for all $w$. Finally, for the set of $w$ where $\mathrm{Var}\left[z|w=w\right]\to 0$, the technical requirement that the entropy maximizing distribution in (2.20) is continuous shows that the entropy difference (B.37) goes to zero in the limit. □

*Proof of Theorem 4:* When $* \in \{\textsc{enc}, \textsc{both}\}$ in (2.23), the encoder can simulate $\mathbf{w}$ by generating it from $(\mathbf{s}, \mathbf{q})$. When $* \in \{\textsc{dec}, \textsc{none}\}$, the encoder can still simulate $\mathbf{w}$ correctly provided that $\mathbf{w}$ and $\mathbf{q}$ are independent. Thus being provided with $\mathbf{w}$ provides no advantage given the conditions of the theorem. □

*Proof of Theorem 5:* We begin by showing

$$R[\text{Q-DEC-W-DEC}](D) = R[\text{Q-NONE-W-DEC}](D). \tag{B.38}$$

When side information $(\mathsf{q}, \mathsf{w})$ is available only at the decoder, the optimal strategy is Wyner-Ziv encoding [190]. Let us compute the optimal reconstruction function $v(\cdot,\cdot,\cdot)$ which maps an auxiliary random variable $\mathsf{u}$ and the side information $\mathsf{q}$ and $\mathsf{w}$ to a reconstruction of the source:

$$\begin{aligned} v(u,q,w) &= \arg\min_{\hat{s}} E[d(\hat{s}, \mathsf{s}; \mathsf{q}) | \mathsf{q} = q, \mathsf{w} = w, \mathsf{u} = u] && \text{(B.39)} \\ &= \arg\min_{\hat{s}} d_0(q) E[d_1(\hat{s}, \mathsf{s}) | \mathsf{q} = q, \mathsf{w} = w, \mathsf{u} = u] && \text{(B.40)} \\ &= \arg\min_{\hat{s}} E[d_1(\hat{s}, \mathsf{s}) | \mathsf{q} = q, \mathsf{w} = w, \mathsf{u} = u] && \text{(B.41)} \\ &= \arg\min_{\hat{s}} E[d_1(\hat{s}, \mathsf{s}) | \mathsf{w} = w, \mathsf{u} = u]. && \text{(B.42)} \end{aligned}$$

We obtain (B.40) from the assumption that we have a separable distortion measure. To get (B.42) recall that by assumption $\mathsf{q}$ is statistically independent of $\mathsf{s}$ given $\mathsf{w}$ and also $\mathsf{q}$ is statistically independent of $\mathsf{u}$ since $\mathsf{u}$ is generated at the encoder from $\mathsf{s}$. Thus neither the optimal reconstruction function, $v(\cdot,\cdot,\cdot)$ nor the auxiliary random variable, $\mathsf{u}$, depend on $\mathsf{q}$. This establishes (B.38).

To show that

$$R[\text{Q-DEC-W-NONE}](D) = R[\text{Q-NONE-W-NONE}](D) \tag{B.43}$$

we need $\mathbf{w}$ and $\mathbf{q}$ to be independent. When this is true, $\mathbf{w}$ does not affect anything and the problem is equivalent to when $\mathbf{w} = 0$ and is available at the decoder. From (B.43) we see that providing $\mathbf{w} = 0$ at the decoder does not help and thus we establish (B.43). Note that this argument fails when $\mathbf{w}$ and $\mathbf{q}$ are not independent since in that case Wyner-Ziv based on $\mathbf{q}$ could be performed and there would be no $\mathbf{w}$ at the decoder to enable the argument in (B.39)–(B.42).

To show that

$$R[\text{Q-DEC-W-BOTH}](D) = R[\text{Q-NONE-W-BOTH}](D) \tag{B.44}$$

we note that in this scenario the encoder and decoder can design a different source coding system for each value of $w$. The subsystem for a fixed value $w^*$ corresponds to source coding with distortion side information at the decoder. Specifically, the source will have distribution $p_{\mathsf{s}|\mathsf{w}}(s|w^*)$, and the distortion side information will have distribution $p_{\mathsf{q}|\mathsf{w}}(q|w^*)$. Thus the performance of each subsystem is given by $R[\text{Q-DEC-W-NONE}](D)$ which we already showed is the same as $R[\text{Q-NONE-W-NONE}](D)$. This establishes (B.44).

Finally, to show that

$$R[\text{Q-DEC-W-ENC}](D) = R[\text{Q-NONE-W-ENC}](D) \tag{B.45}$$

we require the assumption that $\mathbf{q}$ and $\mathbf{w}$ are independent. This assumption implies

$$R[\text{Q-DEC-W-ENC}](D) = R[\text{Q-DEC-W-NONE}](D) \tag{B.46}$$

since an encoder without $\mathbf{w}$ could always generated a simulated $\mathbf{w}$ with the correct distribution relative to the other variables. The same argument implies

$$R[\text{Q-NONE-W-ENC}](D) = R[\text{Q-NONE-W-NONE}](D). \tag{B.47}$$

Combining (B.46), (B.47), and (B.43) yields (B.45). □

*Proof of Theorem 6:* First we establish the four rate-distortion function equalities implied by (2.25a). Using Theorem 3 we have

$$\lim_{D\to D_{\min}} R[\text{Q-ENC-W-DEC}](D) - R[\text{Q-BOTH-W-DEC}](D) \leq \tag{B.48}$$

$$\lim_{D\to D_{\min}} R[\text{Q-ENC-W-DEC}](D) - R[\text{Q-BOTH-W-BOTH}](D) \tag{B.49}$$

$$= 0. \tag{B.50}$$

Similarly,

$$\lim_{D\to D_{\min}} R[\text{Q-ENC-W-BOTH}](D) - R[\text{Q-BOTH-W-BOTH}](D) \leq \tag{B.51}$$

$$\lim_{D\to D_{\min}} R[\text{Q-ENC-W-DEC}](D) - R[\text{Q-BOTH-W-BOTH}](D) \tag{B.52}$$

$$= 0. \tag{B.53}$$

To show that

$$\lim_{D\to D_{\min}} R[\text{Q-ENC-W-NONE}](D) - R[\text{Q-BOTH-W-NONE}](D) = 0 \tag{B.54}$$

we need $\mathbf{q}$ and $\mathbf{w}$ to be independent. When this is true, $\mathbf{w}$ does not affect anything and the problem is equivalent to when $\mathbf{w} = 0$ and is available at the decoder and (B.48)–(B.50) establishes (B.54). Without independence this argument fails because we can no longer invoke Theorem 3 since there will be no $\mathbf{w}$ to make $\mathbf{s}$ and $\mathbf{q}$ conditionally independent in (B.29).

To finish establishing (2.25a) we again require $\mathbf{q}$ and $\mathbf{w}$ to be independent to obtain

$$\lim_{D\to D_{\min}} R[\text{Q-ENC-W-ENC}](D) - R[\text{Q-BOTH-W-ENC}](D) \leq \tag{B.55}$$

$$\lim_{D\to D_{\min}} R[\text{Q-ENC-W-NONE}](D) - R[\text{Q-BOTH-W-ENC}](D) = \tag{B.56}$$

$$\lim_{D\to D_{\min}} R[\text{Q-ENC-W-NONE}](D) - R[\text{Q-BOTH-W-NONE}](D) \tag{B.57}$$

$$= 0 \tag{B.58}$$

where (B.57) follows since the encoder can always simulate $\mathbf{w}$ from $(\mathbf{s}, \mathbf{q})$ and (B.58) follows from (B.54).

Next, we establish the four rate-distortion function equalities implied by (2.25b). Using Theorem 3 we have

$$\lim_{D\to D_{\min}} R[\text{Q-ENC-W-DEC}](D) - R[\text{Q-ENC-W-BOTH}](D) \leq \tag{B.59}$$

$$\lim_{D\to D_{\min}} R[\text{Q-ENC-W-DEC}](D) - R[\text{Q-BOTH-W-BOTH}](D) \tag{B.60}$$

$$= 0. \tag{B.61}$$

Similarly,

$$
\begin{aligned}
\lim_{D \to D_{\min}} R[\textsc{q-both-w-dec}](D) - R[\textsc{q-both-w-both}](D) \leq & \qquad \text{(B.62)} \\
\lim_{D \to D_{\min}} R[\textsc{q-enc-w-dec}](D) - R[\textsc{q-both-w-both}](D) & \qquad \text{(B.63)} \\
= 0. & \qquad \text{(B.64)}
\end{aligned}
$$

To show that

$$\lim_{D \to D_{\min}} R[\textsc{q-none-w-dec}](D) - R[\textsc{q-none-w-both}](D) = 0 \tag{B.65}$$

we need $\mathbf{q}$ and $\mathbf{w}$ to be independent and we need the distortion measure to be of the form $d(\hat{s}, s; q) = d_0(q) \cdot d_1(s - \hat{s})$. When this is true the two rate-distortion functions in (B.65) are equivalent to the Wyner-Ziv rate-distortion function and the conditional rate-distortion function for the difference distortion measure $E[d_0(\mathsf{q})] \cdot d_1(s - \hat{s})$. Thus we can either apply the result from [193] showing these rate-distortion functions are equal in the high-resolution limit or simply specialize Theorem 3 to the case where $\mathbf{q}$ is a constant.

To complete the proof, we again require the assumptions that $\mathbf{q}$ and $\mathbf{w}$ are independent and that the distortion measure is of the form $d(s, \hat{s}; q) = d_0(q) \cdot d_0(q) \cdot d_1(s - \hat{s})$. We have

$$
\begin{aligned}
\lim_{D \to D_{\min}} R[\textsc{q-dec-w-dec}](D) - R[\textsc{q-dec-w-both}](D) \leq & \qquad \text{(B.66)} \\
\lim_{D \to D_{\min}} R[\textsc{q-none-w-dec}](D) - R[\textsc{q-dec-w-both}](D) = & \qquad \text{(B.67)} \\
\lim_{D \to D_{\min}} R[\textsc{q-none-w-dec}](D) - R[\textsc{q-none-w-both}](D) & \qquad \text{(B.68)} \\
= 0. & \qquad \text{(B.69)}
\end{aligned}
$$

where (B.68) follows from Theorem 5 and (B.69) follows from (B.65). □

*Proof of Theorem 7:* We note that according to Theorem 4 and Theorem 6 we can focus solely on the case

$$R[\textsc{q-*-w-none}](D) - R[\textsc{q-*-w-both}](D). \tag{B.70}$$

When * = NONE, the rate difference in (B.70) is the difference between the classical rate-distortion function and the conditional rate-distortion function in the high-resolution limit. Thus the Shannon Lower Bound [109] (and its conditional version) imply that

$$\lim_{D \to D_{\min}} R[\textsc{q-none-w-none}](D) - R[\textsc{q-none-w-both}](D) = h(\mathsf{s}) - h(\mathsf{s}|\mathsf{w}). \tag{B.71}$$

Similarly, when * = DEC an identical argument can be combined with Theorem 5.

When * = ENC, Theorem 6 implies

$$R[\textsc{q-enc-w-none}](D) \to R[\textsc{q-both-w-none}](D)$$

and

$$R[\textsc{q-enc-w-both}](D) \to R[\textsc{q-both-w-both}](D).$$

To find the asymptotic difference between $R[\textsc{q-both-w-none}](D)$ and $R[\textsc{q-both-w-both}](D)$ we note that a separate coding system can be designed for each value of $q$ without loss of optimality. The rate difference for knowledge of $\mathbf{w}$ for each value of $q$ is then given by an expression similar to (B.71) and thus the total rate difference is again $h(s) - h(s|w)$.

Similarly, when $*$ = BOTH an identical argument applies. □

*Proof of Theorem 8:* To simplify the exposition, we first prove the theorem for the relatively simple case of a one-dimensional source ($k = 1$) with a quadratic distortion ($r = 2$). Then at the end of the proof, we describe how to extend it to general $k$ and $r$.

We begin with the case where $*$ = NONE. Since Theorems 5 and 6 imply

$$R[\text{Q-NONE-W-NONE}](D) = R[\text{Q-DEC-W-NONE}](D) \tag{B.72a}$$

and

$$R[\text{Q-ENC-W-NONE}](D) \to R[\text{Q-BOTH-W-NONE}](D) \ , \tag{B.72b}$$

we focus on showing

$$\lim_{D \to D_{\min}} R[\text{Q-BOTH-W-NONE}](D) - R[\text{Q-NONE-W-NONE}](D) = \frac{1}{2} E\left[\ln \frac{E[q]}{q}\right]. \tag{B.73}$$

Computing $R[\text{Q-BOTH-W-NONE}](D)$ is equivalent to finding the rate-distortion function for optimally encoding independent random variables and yields the familiar "water-pouring" rate and distortion allocation [46, Section 13.3.3]. For each $q$, we quantize the corresponding source samples with distortion $D_q = E[(s - \hat{s})^2]$ (or $E[||\mathbf{s} - \hat{\mathbf{s}}||^r]$ in the more general case) and rate $R_q(D_q)$. The overall rate and distortion then become $E[R_q(D_q)]$ and $E[q \cdot D_q]$.

Thus to find the rate and distortion allocation we set up a constrained optimization problem using Lagrange multipliers to obtain the functional

$$J(D) = E[R_q(D_q)] + \lambda(D - E[q \cdot D_q]), \tag{B.74}$$

differentiate with respect to $D_q$, set equal to zero and solve for each $D_q$. In the high-resolution limit, various researchers have shown

$$R_q(D_q) \to h(s) - \frac{1}{2} \log D_q. \tag{B.75}$$

(*e.g.*, see [109] and references therein). Therefore, it is straightforward to show this process yields the condition $D_q = 1/(2\lambda q)$ with $2\lambda = 1/D$ implying

$$\lim_{D \to 0} R[\text{Q-BOTH-W-NONE}](D) \to h(s) - \frac{1}{2} \log D + \frac{1}{2} E[\log q]. \tag{B.76}$$

To compute $R[\text{Q-NONE-W-NONE}](D)$, we note that since neither encoder nor decoder knows $q$ the optimal strategy is to simply quantize the source according to the distortion $d(q, s; \hat{s}) = E[q] \cdot (s - \hat{s})^2$ to obtain

$$\lim_{D \to 0} R[\text{Q-NONE-W-NONE}](D) \to h(s) - \frac{1}{2} \log D + \frac{1}{2} \log E[q]. \tag{B.77}$$

Comparing (B.76) with (B.77) establishes (B.73).

By applying Theorem 4 we see that the case where $*$ = ENC is the same as $*$ = NONE.

Next we consider the case where $*$ = DEC. Since Theorem 6 implies $R[\text{Q-*-W-DEC}](D) =$

$R[\text{Q-*-W-BOTH}](D)$ and $R[\text{Q-ENC-W-*}](D) \to R[\text{Q-BOTH-W-*}](D)$, it suffices to show that

$$\lim_{D\to D_{\min}} R[\text{Q-NONE-W-BOTH}](D) - R[\text{Q-BOTH-W-BOTH}](D) = E\left[\log \frac{E[q]}{q}\right]. \qquad \text{(B.78)}$$

We can compute $R[\text{Q-BOTH-W-BOTH}](D)$ by considering a separate coding system for each value of $w$. Specifically, conditioned on $w = w$, computing the rate-distortion trade-off is equivalent to finding $R[\text{Q-BOTH-W-NONE}](D)$ for a modified source, $s'$ with distribution $p_{s'}(s') = p_{s|w}(s'|w)$. Thus we obtain

$$\lim_{D\to D_{\min}} R[\text{Q-BOTH-W-BOTH}](D) \to h(s|w) - \frac{1}{2}\log D + \frac{1}{2}\log E[q]. \qquad \text{(B.79)}$$

Similarly, we can compute $R[\text{Q-NONE-W-BOTH}](D)$ using an analogous conditioning argument to obtain

$$\lim_{D\to D_{\min}} R[\text{Q-BOTH-W-BOTH}](D) \to h(s|w) - \frac{1}{2}\log D + \frac{1}{2}E[\log q]. \qquad \text{(B.80)}$$

Comparing these two rate-distortion functions yields (B.78).

By applying Theorem 6 we see that the case where * = DEC is the same as * = BOTH.

This establishes the theorem for $k = 1$ and $r = 2$. For general $k$ and $r$, the only change is that each component rate-distortion function $R_q(D_q)$ (B.75) becomes [109, page 2028]

$$R_q(D_q) \to h(s) - \frac{k}{r}\log D_q - \frac{k}{r} + \log\left[\frac{r}{kV_k\Gamma(k/r)}\left(\frac{k}{r}\right)^{k/r}\right]. \qquad \text{(B.81)}$$

and a similar change occurs for all the following rate-distortion expressions. Since we are mainly interested in the difference of rate-distortion functions, most of these extra terms cancel out and the only change is that factors of $1/2$ are replaced with factors of $k/r$. □

## ■ B.4 Finite Resolution Bounds

Before proceeding, we require the following lemma to upper and lower bound the entropy of an arbitrary random variable plus a Gaussian mixture.

**Lemma 9.** *Let $s$ be an arbitrary random variable with finite variance $\sigma^2 < \infty$. Let $z$ be a zero-mean, unit-variance Gaussian independent of $s$ and let $v$ be a random variable independent of $s$ and $z$ with $v_{\min} \le v < v_{\max}$. Then*

$$h(s) + \frac{1}{2}\log(1 + v_{\min}) \le h(s + z\sqrt{v}) \le h(s) + \frac{1}{2}\log(1 + v_{\max}\cdot J(s)) \qquad \text{(B.82)}$$

*with equality if and only if $v$ is a constant and $s$ is Gaussian.*

*Proof.* The concavity of differential entropy yields

$$h(s + z\sqrt{v_{\min}}) \le h(s + z\sqrt{v}) \le h(s + z\sqrt{v_{\max}}). \qquad \text{(B.83)}$$

For the lower bound we have

$$h(s + z\sqrt{v_{\min}}) = \int_0^{v_{\min}} \frac{\delta}{\delta\tau} h(s + z\sqrt{\tau}) d\tau + h(s) \tag{B.84}$$

$$= \int_0^{v_{\min}} \frac{1}{2} J(s + z\sqrt{\tau}) d\tau + h(s) \tag{B.85}$$

$$\geq \frac{1}{2} \int_0^{v_{\min}} J(z\sqrt{\sigma^2 + \tau}) d\tau + h(s) \tag{B.86}$$

$$= \frac{1}{2} \int_0^{v_{\min}} \frac{d\tau}{\sigma^2 + \tau} + h(s) \tag{B.87}$$

$$= \frac{1}{2} \log\left(1 + \frac{v_{\min}}{\sigma^2}\right) \tag{B.88}$$

where (B.85) follows from de Bruijn's identity [46, Theorem 16.6.2], [51, Theorem 14], (B.86) follows from the fact that a Gaussian distribution minimizes Fisher Information subject to a variance constraint, and (B.87) follows since the Fisher Information for a Gaussian is the reciprocal of its variance.

Similarly, for the upper bound we have

$$h(s + z\sqrt{v_{\max}}) = \int_0^{v_{\max}} \frac{\delta}{\delta\tau} h(s + z\sqrt{\tau}) d\tau + h(s) \tag{B.89}$$

$$= \int_0^{v_{\max}} \frac{1}{2} J(s + z\sqrt{\tau}) d\tau + h(s) \tag{B.90}$$

$$\leq \frac{1}{2} \int_0^{v_{\max}} \frac{J(s)J(z\sqrt{\tau})}{J(s) + J(z\sqrt{\tau})} d\tau + h(s) \tag{B.91}$$

$$= \frac{1}{2} \int_0^{v_{\max}} \frac{J(s) d\tau}{\tau J(s) + 1} + h(s) \tag{B.92}$$

$$= \frac{1}{2} \log\left(1 + v_{\max} \cdot J(s)\right) \tag{B.93}$$

where (B.90) again follows from de Bruijn's identity, (B.91) follows from the convolution inequality for Fisher Information [24], [46, p.497], and (B.92) follows since the Fisher Information for a Gaussian is the reciprocal of its variance.

Combining these upper and lower bounds yields the desired result. Finally, the inequalities used in (B.86) and (B.91) are both tight if and only if $s$ is Gaussian. □

As an aside we note that Lemma 9 can be used to bound the rate-distortion function of an arbitrary unit-variance source, $s$, relative to quadratic distortion. Specifically using an additive Gaussian noise test-channel $\hat{s} = z + s$ and combining Lemma 9 to upper bound $h(s + z)$ with the Shannon Lower Bound [109] yields

$$h(s) - \frac{1}{2} \log 2\pi e D \leq R(D) \leq h(s) - \frac{1}{2} \log 2\pi e D + \frac{1}{2} \log[1 + DJ(s)]. \tag{B.94}$$

Evidently, the error in the Shannon Lower Bound is at most $\frac{1}{2} \log[1 + DJ(s)]$ and the sub-optimality of an additive Gaussian noise test-channel is at least $\frac{1}{2} \log[1 + D]$.

*Proof of Theorem 9:* Starting with the bound for the rate gap from (B.29), we have

$$R[\text{Q-ENC-W-DEC}](D) - R[\text{Q-BOTH-W-BOTH}](D) \leq h(s + z|w) - h(s|w) \tag{B.95}$$

$$= \int \left[h(\mathsf{s}+\mathsf{z}|\mathsf{w}=w) - h(\mathsf{s}|\mathsf{w}=w)\right] p_{\mathsf{w}}(w)dw \tag{B.96}$$

$$\leq \int \left\{\frac{1}{2}\log\left(1+\min\left[1, \frac{D}{\mathsf{q}_{\min}}\right]\cdot J(\mathsf{s}|\mathsf{w}=w)\right)\right\} p_{\mathsf{w}}(w)dw \tag{B.97}$$

$$\leq \int \left\{\frac{J(\mathsf{s}|\mathsf{w}=w)}{2}\cdot\min\left[1, \frac{D}{\mathsf{q}_{\min}}\right]\right\} p_{\mathsf{w}}(w)dw \tag{B.98}$$

$$= \frac{J(\mathsf{s}|\mathsf{w})}{2}\cdot\min\left[1, \frac{D}{\mathsf{q}_{\min}}\right]. \tag{B.99}$$

To obtain (B.97) we note that $\mathsf{z}$ is a Gaussian mixture and apply Lemma 9. This follows since, conditioned on $\mathsf{q} = q$, $\mathsf{z}$ is a Gaussian with variance $E[d(\mathsf{s}, \hat{\mathsf{s}}_{\mathsf{w}}^*; q)]$ where $\hat{\mathsf{s}}_{\mathsf{w}}^*$ was defined in the proof of Theorem 3 to be the optimal distribution when both encoder and decoder know the side information. By considering the optimal "water-pouring" distortion allocation for the optimal test-channel distribution $\hat{\mathsf{s}}_{\mathsf{w}}^*$, it can be demonstrated that if the distortion is $D$, then $E[d(\mathsf{s}, \hat{\mathsf{s}}_{\mathsf{w}}^*; q)]$ is at most $\min[1, D/q]$ for each $q$. □

To develop a similar bound for other distortion measures essentially all we need is an upper bound for the derivative of $h(\mathsf{s}+\sqrt{\tau}\mathsf{z})$ with respect to $\tau$. Since entropy is concave, if we can compute this derivative for $\tau = 0$ then it will be an upper bound for the derivative at all $\tau$.

To obtain the desired derivative at $\tau = 0$, we can write

$$h(\mathsf{s}+\sqrt{\tau}\mathsf{z}) = I(\mathsf{s}+\sqrt{\tau}\mathsf{z}; \sqrt{\tau}\mathsf{z}) - h(\mathsf{s}). \tag{B.100}$$

The results of Prelov and van der Meulen [150] imply that under certain regularity conditions

$$\frac{\delta}{\delta\tau}\lim_{\tau\to 0^+} I(\mathsf{s}+\sqrt{\tau}\mathsf{z}; \sqrt{\tau}\mathsf{z}) = J(\mathsf{s})/2 \tag{B.101}$$

which provides the desired derivative. Similarly if we rewrite the mutual information in (B.100) as a relative entropy, then a Taylor series expansion of the relative entropy [94, 2.6] can be used to establish (B.101) provided certain derivatives of the probability distributions exist.

*Proof of Theorem 10:* Starting with the bound for the rate gap from (B.29) in Theorem 3 we have

$$R[\text{Q-ENC-W-DEC}](D) - R[\text{Q-BOTH-W-BOTH}](D) \leq h(\mathsf{s}+\mathsf{z}|\mathsf{w}) - h(\mathsf{s}|\mathsf{w}) \tag{B.102}$$

$$
\begin{aligned}
&= D(\mathsf{s}|\mathsf{w}||\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}})) - D(\mathsf{s}+\mathsf{z}|\mathsf{w}||\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}}+\sigma^2_{\mathsf{z}|\mathsf{w}})) \\
&\quad + h(\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}}+\sigma^2_{\mathsf{z}|\mathsf{w}})|\mathsf{w}) - h(\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}})|\mathsf{w}) && \text{(B.103)} \\
&\leq D(\mathsf{s}|\mathsf{w}||\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}})) + h(\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}}+\sigma^2_{\mathsf{z}|\mathsf{w}})|\mathsf{w}) - h(\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}})|\mathsf{w}) && \text{(B.104)} \\
&= D(\mathsf{s}|\mathsf{w}||\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}})) \\
&\quad + \int \Big[h(\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}}+\sigma^2_{\mathsf{z}|\mathsf{w}})|\mathsf{w}=w) - h(\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}})|\mathsf{w}=w)\Big] p_{\mathsf{w}}(w)dw && \text{(B.105)} \\
&\leq D(\mathsf{s}|\mathsf{w}||\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}})) + \int \left[\frac{1}{2}\log\left(1+\frac{\sigma^2_{\mathsf{z}|\mathsf{w}}}{\sigma^2_{\mathsf{s}|\mathsf{w}}}\right)\right] p_{\mathsf{w}}(w)dw && \text{(B.106)} \\
&\leq D(\mathsf{s}|\mathsf{w}||\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}})) + \int \left[\frac{1}{2}\log\left(1+\frac{\sigma^2_{\max}}{\sigma^2_{\mathsf{s}|\mathsf{w}}}\right)\right] p_{\mathsf{w}}(w)dw && \text{(B.107)} \\
&\leq D(\mathsf{s}|\mathsf{w}||\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}})) + \int \left[\frac{1}{2}\log\left(1+\frac{\sigma^2_{\max}}{\sigma^2_{\min}}\right)\right] p_{\mathsf{w}}(w)dw && \text{(B.108)} \\
&= D(\mathsf{s}|\mathsf{w}||\mathcal{N}(\sigma^2_{\mathsf{s}|\mathsf{w}})) + \frac{1}{2}\log\left(1+\frac{\sigma^2_{\max}}{\sigma^2_{\min}}\right). && \text{(B.109)}
\end{aligned}
$$

To obtain (B.103)–(B.106) we use the same arguments as in (B.31)–(B.37) plus the additional observation that relative entropy is positive and can be dropped in obtaining (B.104). Next, we note that there is never any need to choose the variance of the test-channel noise to be larger than $\sigma^2_{\max}$ to get (B.107). Finally, the assumption that $\sigma^2_{\mathsf{s}|\mathsf{w}} \geq \sigma^2_{\min}$ yields (B.108). □

To develop a similar bound for other distortion measures, we would use an entropy maximizing distribution for the appropriate distortion measure in $D(\mathsf{s}|\mathsf{w}||\cdot)$ and $D(\mathsf{s}+\mathsf{z}||\mathsf{w}||\cdot)$ above.

Appendix C

# Iterative Quantization Proofs

*Proof of Theorem 14:* Consider quantizing a random source and choose some $c > 0$ and let $d$ be the smallest integer such that at least $c{\cdot}n$ parity checks have degree at most $d$. For each such parity check, the probability that all variables in the check are not erased is at least $(1-e)^d$. Hence the probability that the check cannot be satisfied is at least $(1/2)\cdot(1-e)^d$. Since there are $c \cdot n$ such checks, the probability that at least one check cannot be satisfied is

$$\begin{aligned}
\Pr[\text{encoding failure}] &\geq 1 - \left[1 - (1/2)\cdot(1-e)^d\right]^{c\cdot n} && \text{(C.1)}\\
&\geq 1 - \left[1 - (1/2 - e/2)^d\right]^{c\cdot n} && \text{(C.2)}\\
&= 1 - \exp\left\{c\cdot n \ln\left[1 - (1/2 - e/2)^d\right]\right\} && \text{(C.3)}\\
&\geq 1 - \exp\left\{-c\cdot n \cdot (1/2 - e/2)^d\right\} && \text{(C.4)}\\
&= 1 - \exp\left\{-\exp\left[\ln c + \ln n + d \ln\left(1/2 - e/2\right)\right]\right\}. && \text{(C.5)}
\end{aligned}$$

Hence for the probability of decoding failure to become small, $d$ must grow at least logarithmically with $n$ for every $c > 0$. Note that this argument applies to any parity-check graph representation of the code. □

*Proof of Theorem 15:* We will show that unique channel decoding is possible if and only if the matrix equation $\mathbf{Mx} = \mathbf{y}$ has a solution (where $\mathbf{M}$ will be defined shortly). Similarly, we will show that source quantization is possible for every $\mathbf{z}$ if and only if the matrix equation $\mathbf{wM} = \mathbf{z}$ has a solution for every $\mathbf{z}$. By demonstrating that both conditions are satisfied if and only if the same matrix $\mathbf{M}$ has rank $n$, we will prove the desired result.

Assume that all erasures occur in the last $|\mathbf{e}|$ positions (*i.e.*, $\mathbf{e} = 0^{n-|\mathbf{e}|}\ 1^{|\mathbf{e}|}$). [1] This incurs no loss of generality since the coordinates of $\mathcal{C}$ can always be permuted accordingly and the theorem applied to the permuted code and its permuted dual code. Let $\mathbf{x}$ represent the transmitted signal and let $\mathbf{y}$ denote the received signal. Optimal decoding corresponds to finding a vector which is a codeword of $\mathcal{C}$ and consistent with the unerased received values. The requirement that $\mathbf{x}$ is a codeword corresponds to the equation $\mathbf{Hx} = 0$ where $\mathbf{H}$ is the parity check matrix of $\mathcal{C}$. The requirement that $\mathbf{x}$ is consistent with the received unerased data corresponds to the equation $(\mathbf{I}_{n-|\mathbf{e}|}\ \mathbf{0})\mathbf{x} = \mathbf{y}\,_1^{n-|\mathbf{e}|}$ where $\mathbf{I}_t$ represents a $t$-by-$t$ identity matrix and $\mathbf{y}\,_i^j$ represents the sub-vector $(y_i, y_{i+1}, \ldots, y_j)$. Thus successful decoding is possible if and only if the equation

$$\begin{pmatrix} \mathbf{I}_{n-|\mathbf{e}|} & \mathbf{0} \\ \multicolumn{2}{c}{\mathbf{H}} \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mathbf{y}\,_1^{n-|\mathbf{e}|} \\ 0 \end{pmatrix} \tag{C.6}$$

has a unique solution. According to well-known properties of linear algebra, uniqueness is equivalent to the matrix in (C.6) having full column rank (*i.e.*, rank $n$). Note that existence of a solution is guaranteed since a codeword was sent and no errors occurred.

Let $\mathbf{z}$ represent the source to be quantized with erasure pattern $\mathbf{e}^{\perp}$. Since we assumed that all erasures in $\mathbf{e}$ occurred in the last $|\mathbf{e}|$ samples, the dual erasure pattern $\mathbf{e}^{\perp}$ has all erasures occurring in the first $|\mathbf{e}^{\perp}|$ positions (*i.e.*, $\mathbf{e}^{\perp} = 1^{n-|\mathbf{e}|}\ 0^{|\mathbf{e}|}$). Optimal decoding corresponds to finding a vector $\mathbf{w}$ which is a codeword of $\mathcal{C}$ and consistent with the unerased

[1] We use $|\mathbf{a}|$ to denote the number of non-zero values in $\mathbf{a}$ (*i.e.*, the weight of $\mathbf{a}$) and $b^c \triangleq \underbrace{(b\ b \ldots b)}_{\text{c times}}$.

received values. The former requirement corresponds to the equation $\mathbf{v}\mathbf{G}^{\perp} = \mathbf{w}$ where $\mathbf{G}^{\perp}$ is the generator matrix of $\mathcal{C}^{\perp}$ and $\mathbf{v}$ is a binary vector of appropriate dimension. The latter requirement corresponds to the equation $\mathbf{u}\begin{pmatrix} \mathbf{I}_{|\mathbf{e}^{\perp}|} & \mathbf{0} \end{pmatrix} + \mathbf{w} = \mathbf{z}$ where $\mathbf{u}$ is a binary vector chosen to ensure that the first $|\mathbf{e}^{\perp}|$ positions (*i.e.*, the erased positions) match regardless of $\mathbf{w}$. Thus successful decoding is possible for every $\mathbf{z}$ if and only if a solution exists for

$$\begin{pmatrix} \mathbf{u} & \mathbf{v} \end{pmatrix} \begin{pmatrix} \mathbf{I}_{|\mathbf{e}^{\perp}|} & \mathbf{0} \\ \multicolumn{2}{c}{\mathbf{G}^{\perp}} \end{pmatrix} = \mathbf{z} \tag{C.7}$$

for every $\mathbf{z}$. According to well-known properties of linear algebra, existence of a solution for every $\mathbf{z}$ is equivalent to the matrix in (C.7) having full column rank (*i.e.*, rank $n$). Note that uniqueness of a solution is neither guaranteed not required since quantization is successful if at least one solution is found.

Noting that $\mathbf{G}^{\perp} = \mathbf{H}$ and $|\mathbf{e}^{\perp}| = n - |\mathbf{e}|$ completes the proof since these conditions imply that the matrices in (C.6) and (C.7) are identical. □

*Proof of Theorem 16:* For the algorithm to exit the while loop and reach step 8, every unerased element of $z_i$ must have been erased in step 3 and assigned a reserved message variable. After a variable is reserved all its checks must be erased. Since checks can never changed from erased to unerased, a reserved variable can never again be selected in step 2 and thus a variable can never be reserved more than once.

Thus after the while loop, each unerased position in $\mathbf{z}$ has a corresponding reserved variable. Hence there exists an assignment of the message variables which result in a codeword matching $\mathbf{z}$ in the unerased positions. This assignment could be computed via brute-force by solving he corresponding system of linear equations, but in Theorem 18 we show that this step can be computed more efficiently. □

*Proof of Theorem 17:* The proof relies on the following invariant for steps 1–7 of both algorithms:

$$\forall j,\ y_j \text{ is erased if and only if } z_j \text{ is unerased.} \tag{C.8}$$

This condition is trivially true before the algorithm begins and forms the base case for a proof by induction. We assume that (C.8) holds at iteration $i$ of steps 1–7 and show that it must also hold at iteration $i + 1$.

First, (C.8) implies that the outcome of step 1 is the same for each algorithm. Next, since $\mathbf{G}^{\perp} = \mathbf{H}$ the tests in step 1 and step 2 of ERASURE-DECODE($\mathbf{H}, \mathbf{y}$) and ERASURE-QUANTIZE($\mathbf{G}^{\perp}, \mathbf{z}$) yield the same result. Finally, at step 3, $y_j$ is unerased while $z_j$ is erased. Therefore, by induction, condition (C.8) is true at every iteration and ERASURE-DECODE($\mathbf{H}, \mathbf{y}$) fails at step 5 if and only if ERASURE-QUANTIZE($\mathbf{G}^{\perp}, \mathbf{z}$) fails at step 5. □

*Proof of Theorem 18:* The while loop executes at most $n$ times. Therefore step 1 requires at most $\mathcal{O}(n)$ operations. Consider storing the number of variables with exactly one unerased check in a data structure which supports insertion and removal in constant time (*e.g.*, a hash table). We can initialize the data structure with $\mathcal{O}(d \cdot n)$ operations. Removing an element in steps 2 and 3 and updating the data structure to account for step 3 requires $\mathcal{O}(d)$ operations. Thus steps 1 through 8 require $\mathcal{O}(d \cdot n)$ operations and all that remains is to bound the running time of step 9.

Denote the first reserved variable by $v_{j(1)}$, the second reserved variable by $v_{j(2)}$ and so on to $v_{j(|e|)}$. As described in step 9, we first assign a value to $v_{j(|e|)}$ and work backward.

Specifically, we set $v_{j(i)}$ to the modulo-2 sum of $z_{j(i)}$ and all message variables connected to $z_{j(i)}$ (except $v_{j(i)}$). This is possible for $z_{j(|e|)}$ since no other reserved variable could be connected to $z_{j(|e|)}$.[2] Similarly, $z_{j(|e|-1)}$ must be connected to only unreserved variables as well as perhaps to $v_{j(|e|)}$ and therefore a value can be determined for $v_{j(|e|-1)}$. Thus, by induction we can determine every $v_j$.

Adding up the operations computed for each step yields a running time of $\mathcal{O}(d \cdot n)$.

□

[2] If $z_{j(|e|)}$ was connected to another reserved variable $v$, that would imply $v$ was reserved when connected to $z_{j(|e|)}$ which was unerased as well as $z$ which must also have been unerased. This contradicts step 2 in ERASURE-QUANTIZE.

Appendix D

# Information/Operational R(D) Equivalence

**Proposition 3.** *The information/operational feedforward rate-distortion functions are equal.*

*Proof.* Since the decoder must deterministically produce $\hat{\mathsf{s}}[i]$ from $\mathsf{s}[^{i-1}_{1}]$ and the $nR$ bits produced by the encoder we have

$$
\begin{aligned}
nR &\geq \sum_{i=1}^{n} H(\hat{\mathsf{s}}[i]\,|\mathsf{s}[^{i-1}_{1}]) && \text{(D.1)}\\
&\geq \sum_{i=1}^{n} [H(\hat{\mathsf{s}}[i]\,|\mathsf{s}[^{i-1}_{1}]) - H(\hat{\mathsf{s}}[i]\,|\mathsf{s}[^{i}_{1}])] && \text{(D.2)}\\
&= \sum_{i=1}^{n} [H(\mathsf{s}[i]\,|\mathsf{s}[^{i-1}_{1}]) - H(\mathsf{s}[i]\,|\mathsf{s}[^{i-1}_{1}], \hat{\mathsf{s}}[i])] && \text{(D.3)}\\
&= \sum_{i=1}^{n} [H(\mathsf{s}[i]) - H(\mathsf{s}[i]\,|\mathsf{s}[^{i-1}_{1}], \hat{\mathsf{s}}[i])] && \text{(D.4)}\\
&\geq \sum_{i=1}^{n} [H(\mathsf{s}[i]) - H(\mathsf{s}[i]\,|\hat{\mathsf{s}}[i])] && \text{(D.5)}
\end{aligned}
$$

where (D.4) follows since the source is memoryless and (D.5) follows since conditioning reduces entropy. From this point standard convexity arguments establish that (5.1) is a lower bound to the average rate. □

Appendix E

# Proofs For Burst-Delay Codes

*Proof of Lemma 2:* Based on (7.20) there are at most

$$2^{n_c \cdot R/\mathtt{M}} \cdot \left(2^{n_c \cdot [R/\mathtt{M} - \mathtt{M} \cdot I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})]/(\mathtt{L}-1)}\right)^{\mathtt{B}} \tag{E.1}$$

possible values for the unknown source components. The correct combination will always be jointly typical with the bin index. The probability that a particular incorrect combination of source components is typical is exactly the inverse of the number of possible bin indexes:

$$2^{-n_c \cdot [I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}})-\epsilon]}. \tag{E.2}$$

Thus the probability that any incorrect combination is jointly typical is at most

$$2^{n_c \cdot R/\mathtt{M}} \cdot \left(2^{n_c \cdot [R/\mathtt{M} - I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})]/(\mathtt{L}-1)}\right)^{\mathtt{B}} \cdot 2^{-n_c \cdot [I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}})-\epsilon]}. \tag{E.3}$$

Provided $R < C_{\mathtt{B},\mathtt{T}} - \mathtt{M} \cdot \epsilon$, the exponent is strictly less than

$$n_c \cdot \left[\frac{C_{\mathtt{B},\mathtt{T}} - \epsilon \mathtt{M}}{\mathtt{M}} + (C_{\mathtt{B},\mathtt{T}} - \mathtt{M}\epsilon) \cdot \frac{\mathtt{B}}{(\mathtt{L}-1)\mathtt{M}} - \frac{\mathtt{B}}{\mathtt{L}-1} \cdot I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}}) - I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}}) + \epsilon\right] \tag{E.4}$$

$$\begin{aligned}
&= n_c \Bigg\{ -\frac{\mathtt{B}\epsilon}{\mathtt{L}-1} + I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - \frac{\mathtt{B}}{\mathtt{M}(\mathtt{B}+\mathtt{T})} \cdot [I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})] \\
&+ \frac{\mathtt{B}}{\mathtt{L}-1} \cdot [I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})] \\
&- \frac{\mathtt{B}^2}{(\mathtt{B}+\mathtt{T})\mathtt{M}(\mathtt{L}-1)} \cdot [I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})] - I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}}) \Bigg\} &&\text{(E.5)} \\
&= n_c \Bigg\{ -\frac{\mathtt{B}\epsilon}{\mathtt{L}-1} \\
&+ [I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})] \cdot \left[\frac{\mathtt{B}}{\mathtt{L}-1} - \frac{\mathtt{B}}{\mathtt{M}(\mathtt{T}+\mathtt{B})} - \frac{\mathtt{B}^2}{\mathtt{M}(\mathtt{T}+\mathtt{B})(\mathtt{L}-1)}\right] &&\text{(E.6)} \\
&= n_c \Bigg\{ -\frac{\mathtt{B}\epsilon}{\mathtt{L}-1} \\
&+ \mathtt{B} \cdot \frac{I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{G}}) - I(x;y|\boldsymbol{\theta}=\theta_{\mathcal{B}})}{\mathtt{M}(\mathtt{T}+\mathtt{B})(\mathtt{L}-1)} \cdot [\mathtt{M}(\mathtt{T}+\mathtt{B}) - (\mathtt{L}-1) - \mathtt{B}] \Bigg\} &&\text{(E.7)} \\
&= -n_c \frac{\mathtt{B}\epsilon}{\mathtt{L}-1}. &&\text{(E.8)}
\end{aligned}$$

Thus for $R < C_{\mathtt{B},\mathtt{T}} - \mathtt{M} \cdot \epsilon$, the probability of error is at most

$$2^{-n_c \frac{\mathtt{B}\epsilon}{\mathtt{L}-1}} \tag{E.9}$$

which goes to zero asymptotically with $n_c$.

□

Appendix F

# Proofs for Burst Correcting Code Constructions

*Proof of Theorem 24:* Imagine that a convolutional packet code is constructed from a block code $\mathbf{C}$ using the structure in Fig. 8-2 and consider a burst of length $\mathtt{B}$ starting at time $i$. Since the resulting convolutional packet code is time invariant, showing that $\mathbf{s}[i]$ can be recovered by time $i + \mathtt{T}$ is sufficient to establish the desired result. We do this by separately considering the $k$ source components $s_j[i]$ with $j \in \{0, 1, \ldots, k-1\}$. Each such source component is the $j$th component in a different "diagonal block code" as illustrated in Fig. F-1.

In general, $s_j[i]$ is the $j$th symbol in $\mathbf{w}[i + k - 1 - j]$. The burst starting at $\mathbf{x}[i]$ affects $\mathbf{w}_j[i+k-1-j]$ through $\mathbf{w}_{j+\mathtt{B}-1}[i+k-1-j]$ and can be recovered once $\mathbf{w}_{j+\mathtt{B}}[i+k-1-j]$ through $\mathbf{w}_{j+\mathtt{T}}[i+k-1-j]$ are received by the assumption about the delay $\mathbf{C}$ requires to correct a burst of length $\mathtt{B}$. Since $\mathbf{w}_{j+\mathtt{T}}[i+k-1-j]$ is an element of $\mathbf{x}[i+\mathtt{T}]$, we have that $s_j[i]$ is recovered with delay $\mathtt{T}$.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\mathbf{x}[i-2] =$ | ( | $s_0[i-2]$ | $s_1[i-2]$ | $s_2[i-2]$ | $s_0[i-5] \oplus s_2[i-3]$ | $s_1[i-5] \oplus s_2[i-4]$ | ) |
| $\mathbf{x}[i-1] =$ | ( | $s_0[i-1]$ | $s_1[i-1]$ | $s_2[i-1]$ | $s_0[i-4] \oplus s_2[i-2]$ | $s_1[i-4] \oplus s_2[i-3]$ | ) |
| ~~$\mathbf{x}[i] =$~~ | ~~(~~ | ~~$\boxed{s_0[i]}$~~ | ~~$s_1[i]$~~ | ~~$s_2[i]$~~ | ~~$s_0[i-3] \oplus s_2[i-1]$~~ | ~~$s_1[i-3] \oplus s_2[i-2]$~~ | ~~)~~ |
| ~~$\mathbf{x}[i+1] =$~~ | ~~(~~ | ~~$s_0[i+1]$~~ | ~~$\boxed{s_1[i+1]}$~~ | ~~$s_2[i+1]$~~ | ~~$s_0[i-2] \oplus s_2[i]$~~ | ~~$s_1[i-2] \oplus s_2[i-1]$~~ | ~~)~~ |
| $\mathbf{x}[i+2] =$ | ( | $s_0[i+2]$ | $s_1[i+2]$ | $\boxed{s_2[i+2]}$ | $s_0[i-1] \oplus s_2[i+1]$ | $s_1[i-1] \oplus s_2[i]$ | ) |
| $\mathbf{x}[i+3] =$ | ( | $s_0[i+3]$ | $s_1[i+3]$ | $s_2[i+3]$ | $\boxed{s_0[i] \oplus s_2[i+2]}$ | $s_1[i] \oplus s_2[i+1]$ | ) |
| $\mathbf{x}[i+4] =$ | ( | $s_0[i+4]$ | $s_1[i+4]$ | $s_2[i+4]$ | $s_0[i+1] \oplus s_2[i+3]$ | $\boxed{s_1[i+1] \oplus s_2[i+2]}$ | ) |

(a) A burst starting at time $i$ affects the first two symbols of the constituent diagonal block code starting with $s_0[i]$.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\mathbf{x}[i-2] =$ | ( | $s_0[i-2]$ | $s_1[i-2]$ | $s_2[i-2]$ | $s_0[i-5] \oplus s_2[i-3]$ | $s_1[i-5] \oplus s_2[i-4]$ | ) |
| $\mathbf{x}[i-1] =$ | ( | $\boxed{s_0[i-1]}$ | $s_1[i-1]$ | $s_2[i-1]$ | $s_0[i-4] \oplus s_2[i-2]$ | $s_1[i-4] \oplus s_2[i-3]$ | ) |
| ~~$\mathbf{x}[i] =$~~ | ~~(~~ | ~~$s_0[i]$~~ | ~~$\boxed{s_1[i]}$~~ | ~~$s_2[i]$~~ | ~~$s_0[i-3] \oplus s_2[i-1]$~~ | ~~$s_1[i-3] \oplus s_2[i-2]$~~ | ~~)~~ |
| ~~$\mathbf{x}[i+1] =$~~ | ~~(~~ | ~~$s_0[i+1]$~~ | ~~$s_1[i+1]$~~ | ~~$\boxed{s_2[i+1]}$~~ | ~~$s_0[i-2] \oplus s_2[i]$~~ | ~~$s_1[i-2] \oplus s_2[i-1]$~~ | ~~)~~ |
| $\mathbf{x}[i+2] =$ | ( | $s_0[i+2]$ | $s_1[i+2]$ | $s_2[i+2]$ | $\boxed{s_0[i-1] \oplus s_2[i+1]}$ | $s_1[i-1] \oplus s_2[i]$ | ) |
| $\mathbf{x}[i+3] =$ | ( | $s_0[i+3]$ | $s_1[i+3]$ | $s_2[i+3]$ | $s_0[i] \oplus s_2[i+2]$ | $\boxed{s_1[i] \oplus s_2[i+1]}$ | ) |
| $\mathbf{x}[i+4] =$ | ( | $s_0[i+4]$ | $s_1[i+4]$ | $s_2[i+4]$ | $s_0[i+1] \oplus s_2[i+3]$ | $s_1[i+1] \oplus s_2[i+2]$ | ) |

(b) A burst starting at time $i$ affects the second and third symbols of the constituent diagonal block code starting with $s_0[i-1]$.

**Figure F-1.** Illustration of how a burst on a rate 3/5 convolutinal code affects the constituent interleaved block codes.

For example, as illustrated in Fig. F-1(a), $s_0[i]$ corresponds to $v_2[i+k-1]$ in Fig. 8-2 and is encoded with the block

$$\mathbf{v}[i+k-1] = (s_0[i], s_1[i+1], s_2[i+2], \ldots, s_{k-1}[i+k-1]) \tag{F.1}$$

to yield $\mathbf{w}[i+k-1]$. The burst erases $x[i]$, $x[i+1]$, …, $x[i+\mathtt{B}-1]$, or equivalently the first $\mathtt{B}$ symbols of the constituent block code (*i.e.*, $v_2[i+k-1]$, $v_2[i+k-1]$, …, $v_2[i+k-1]$). Thus by the assumption about the delay $\mathbf{C}$ requires, $v_2[i+k-1]$ can be recovered once $v_2[i+k-1]$, $v_2[i+k-1]$, …, $v_2[i+k-1]$ are received. Thus, $s_0[i]$ can be recovered once the unerased packets up to and including $\mathbf{s}[i+\mathtt{T}]$ are received and $s_0[i]$ is recovered with the appropriate delay.

A similar argument follows for $s_1[i]$ as shown in Fig. F-1(b). Specifically, $s_1[i]$ corresponds to $v_2[i+k-2]$ in Fig. 8-2 and is encoded with the block

$$\mathbf{v}[i+k-2] = (s_1[i], s_2[i+1], s_3[i+2], \ldots, s_{k-1}[i+k-2]) \tag{F.2}$$

to yield $\mathbf{w}[i+k-2]$. The burst erases $v_2[i+k-2]$, $v_2[i+k-2]$, …, $v_2[i+k-2]$. Thus by the assumption about the delay $\mathbf{C}$ requires, $v_2[i+k-2]$ can be recovered once $v_2[i+k-2]$, $v_2[i+k-2]$, …, $v_2[i+k-2]$ are received. Thus, $s_1[i]$ can be recovered once the unerased packets up to and including $\mathbf{s}[i+\mathtt{T}]$ are received and $s_1[i]$ is recovered with the appropriate delay.

Thus we see that in general $s_j[i]$ is part of a constituent block code with the burst affecting symbols $j$ through $j+\mathtt{B}-1$ of the block code. By our assumption about the burst correcting capabilities of the block code, such bursts can be corrected with the appropriate delay. So the burst correcting abilities of the block code translate into similar abilities for the convolutional code.

□

*Proof of Theorem 26:* First imagine that a burst of length $\mathtt{B} = n-k$ occurs on the second stream in the code construction in Fig. 8-5 (*i.e.*, the burst affects $x_{2,j}$). Since the entire first stream is received correctly, its effect on the second stream can be computed and subtracted out. Thus any burst of length $n-k$ on the second stream can be decoded with the required delay by the argument used for a single link code in Theorem 25.

Next, consider a burst of length $\mathtt{B} = n-k$ occurring at time $i$ on the first stream (*i.e.*, $x_{1,i}$, $x_{1,i+1}$, …, $x_{1,i+\mathtt{B}-1}$). Since the source symbols for the second stream are systematically encoded, the corresponding parity symbols can be computed and subtracted out from $x_{2,j}$. Thus the decoder effectively receives the unerased symbols from the first stream as well as the linear combination of the parity check symbols output from the top single-link code in Fig. 8-5 with $s_{1,k-n/2}$, $s_{1,k-n/2+1}$, …, $s_{1,n/2-1}$. The unerased symbols in $s_{1,k-n/2}$, $s_{1,k-n/2+1}$, …, $s_{1,n/2-1}$ can be subtracted out of this linear combination.

After these two subtractions, the decoder effectively receives the output of a single-link code with input $s_{1,0}$, $s_{1,1}$, …, $s_{1,k-n/2-1}$ except for $\mathtt{B}$ "virtually erased" symbols starting at time $i$. We use the term virtually erased since these symbols are not truly erased, but received linearly combined with erased symbols and thus not useful. In any case, the loss of the virtually erased symbols can be corrected with the required delay using the arguments from Theorem 25.

Finally, once $s_{1,0}$, $s_{1,1}$, …, $s_{1,k-n/2-1}$ are decoded, the corresponding parity check symbols for the output of the top single-link encoder in Fig. 8-5 can be computed and subtracted from the second stream. At this point the second stream contains $s_{1,k-n/2}$, $s_{1,k-n/2+1}$, …, $s_{1,n/2-1}$ and thus these symbols are recovered with the appropriate delay. □

*Proof of Theorem 27:* Imagine that a burst occurs on the last stream. Then the data for all the other streams is correctly received and can be subtracted off. Thus since the single-link code for the last stream is optimal, the burst can be corrected with the required delay. If a burst occurs on any stream $j < \mathtt{M}$, then the data for all streams except $j$ and the last stream are received and can be subtracted off. The argument to show that the remaining two streams can be correctly recovered is then the same as for Theorem 26. □

Appendix G

# Proofs For Delay Universal Streaming Codes

*Proof of Lemma 6:* We have the following chain of inequalities:

$$
\begin{aligned}
N \cdot n_c \cdot \mathtt{R} &< \mathsf{I}_d[i_f, \mathtt{R}|\boldsymbol{\theta}] && \text{(G.1)}\\
&= \sum_{i=i_s}^{i_f} \left[n_c\mathtt{R} - I(\mathbf{x}\,[i]\,; \mathbf{y}\,[i]\,|\boldsymbol{\theta}\,[i])\right] && \text{(G.2)}\\
&= (i_f - i_s + 1)n_c\mathtt{R} - I\left(\mathbf{x}\left[{}^{i_f}_{i_s}\right]; \mathbf{y}\left[{}^{i_f}_{i_s}\right]|\boldsymbol{\theta}\left[{}^{i_f}_{i_s}\right]\right) && \text{(G.3)}\\
&= (i_f - i_s + 1)n_c\mathtt{R} - I\left(\mathbf{x}\left[{}^{i_f}_{i_s}\right]; \mathbf{y}\left[{}^{i_f}_{-\infty}\right]|\boldsymbol{\theta}\left[{}^{i_f}_{-\infty}\right]\right) && \text{(G.4)}\\
&\le (i_f - i_s + 1)n_c\mathtt{R} - I\left(\mathbf{s}\left[{}^{i_f}_{i_s}\right]; \mathbf{y}\left[{}^{i_f}_{-\infty}\right]|\boldsymbol{\theta}\left[{}^{i_f}_{-\infty}\right]\right) && \text{(G.5)}\\
&\le (i_f - i_s + 1)n_c\mathtt{R} - I\left(\mathbf{s}\left[{}^{i_f-N}_{i_s}\right]; \mathbf{y}\left[{}^{i_f}_{-\infty}\right]|\boldsymbol{\theta}\left[{}^{i_f}_{-\infty}\right]\right) && \text{(G.6)}\\
I\left(\mathbf{s}\left[{}^{i_f-N}_{i_s}\right]; \mathbf{y}\left[{}^{i_f}_{-\infty}\right]|\boldsymbol{\theta}\left[{}^{i_f}_{-\infty}\right]\right) &< (i_f - i_s - N + 1)n_c\mathtt{R}. && \text{(G.7)}
\end{aligned}
$$

Equations (G.1) and (G.2) follow from the assumption that $\mathsf{I}_d[i, \mathtt{R}|\boldsymbol{\theta}] > n_c\mathrm{R}N$ for $i_s \le i \le i_f$. The chain rule for mutual information and the block memoryless structure of the channel yield (G.3) and (G.4). Equation (G.5) follows from the data processing inequality and (G.6) follows from the chain rule for mutual information.

Finally, we can consider $\mathbf{s}\,[i_s]$, $\mathbf{s}\,[i_s+1]$, …, $\mathbf{s}\,[i_f - N]$ as a message of rate $(i_f - i_s - N + 1) \cdot \mathtt{R}$. Therefore Fano's inequality combined with (G.7) implies that the decoder can not recover this message with small probability of error. □

*Proof of Lemma 7:* Consider the greedy channel state sequence $\boldsymbol{\theta}^*$ where first $\theta^*\,[i_s]$ is chosen to maximize the information debt when it becomes positive at time $i_s$, next $\theta^*\,[i_s+1]$ is chosen to maximize $\mathsf{I}_d[i_s+1, \mathtt{R}|\boldsymbol{\theta}^*]$, and then $\theta^*\,[i_s+2]$ is chosen to maximize $\mathsf{I}_d[i_s+2, \mathtt{R}|\boldsymbol{\theta}^*]$, *etc*, until $\mathsf{I}_d[i, \mathtt{R}|\boldsymbol{\theta}^*]$ becomes non-positive at time $i_s + \mathtt{T}$ and stays non-positive for at least the next $\mathtt{T}$ times.[1] By construction, this choice for $\theta^*\,[i]$ maximizes the right hand side of (9.10).

Without loss of generality, let $i_s = 0$, and assume, for the sake of contradiction, that some code, $\mathbf{C}$, achieves a lower delay than (9.10). Then, with this code, $\mathbf{s}\,[0]$, can be decoded at or before time $\mathtt{T}-1$ when $\mathsf{I}_d[\mathtt{T}-1, \mathtt{R}|\boldsymbol{\theta}^*] = \delta > 0$. Define the new channel state sequences $\boldsymbol{\theta}^{**}$ and $\boldsymbol{\theta}^{**'}$ by letting $\boldsymbol{\theta}^{**} = \boldsymbol{\theta}^*$, $\boldsymbol{\theta}^{**'} = \boldsymbol{\theta}^*$ except that $\theta^{**'}\,[\mathtt{T}] = \theta^*\,[0]$, $\theta^{**}\,[\mathtt{T}] = \theta^*\,[0]$, and $\theta^{**}\,[0] = \theta^*\,[\mathtt{T}]$. The relationships between these channel state sequences and others we define shortly are illustrated in Table G.1.

**Table G.1.** The relationship between various channel state sequences used in the proof. Sequences without the prime symbol ($'$) are elements of $\boldsymbol{\Theta}$, while those with primes may not be in $\boldsymbol{\Theta}$, but must be decodable with delay no more than the corresponding unprimed sequence.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\boldsymbol{\theta}^*$ | $=$ | $\theta^*\,[-1]$ | $\theta^*\,[0]$ | $\theta^*\,[1]$ | $\theta^*\,[2]$ | $\ldots$ | $\theta^*\,[\mathtt{T}-1]$ | $\theta^*\,[\mathtt{T}]$ | $\theta^*\,[\mathtt{T}+1]$ | $\theta^*\,[\mathtt{T}+2]$ | $\theta^*\,[\mathtt{T}+3]$ |
| $\boldsymbol{\theta}^{**}$ | $=$ | $\theta^*\,[-1]$ | $\theta^*\,[\mathtt{T}]$ | $\theta^*\,[1]$ | $\theta^*\,[2]$ | $\ldots$ | $\theta^*\,[\mathtt{T}-1]$ | $\theta^*\,[0]$ | $\theta^*\,[\mathtt{T}+1]$ | $\theta^*\,[\mathtt{T}+2]$ | $\theta^*\,[\mathtt{T}+3]$ |
| $\boldsymbol{\theta}^{**'}$ | $=$ | $\theta^*\,[-1]$ | $\theta^*\,[0]$ | $\theta^*\,[1]$ | $\theta^*\,[2]$ | $\ldots$ | $\theta^*\,[\mathtt{T}-1]$ | $\theta^*\,[0]$ | $\theta^*\,[\mathtt{T}+1]$ | $\theta^*\,[\mathtt{T}+2]$ | $\theta^*\,[\mathtt{T}+3]$ |
| $\boldsymbol{\theta}^{***}$ | $=$ | $\theta^*\,[-1]$ | $\theta^*\,[\mathtt{T}]$ | $\theta^*\,[\mathtt{T}+1]$ | $\theta^*\,[2]$ | $\ldots$ | $\theta^*\,[\mathtt{T}-1]$ | $\theta^*\,[0]$ | $\theta^*\,[1]$ | $\theta^*\,[\mathtt{T}+2]$ | $\theta^*\,[\mathtt{T}+3]$ |
| $\boldsymbol{\theta}^{***'}$ | $=$ | $\theta^*\,[-1]$ | $\theta^*\,[0]$ | $\theta^*\,[1]$ | $\theta^*\,[2]$ | $\ldots$ | $\theta^*\,[\mathtt{T}-1]$ | $\theta^*\,[0]$ | $\theta^*\,[1]$ | $\theta^*\,[\mathtt{T}+2]$ | $\theta^*\,[\mathtt{T}+3]$ |

[1] If it is not possible to choose $\boldsymbol{\theta}$ in a permutation invariant set such that $\mathsf{I}_d[i, \mathtt{R}|\boldsymbol{\theta}]$ stays non-positive for at least $\mathtt{T}$ times, then it is possible to show that $\mathsf{I}_d[i, \mathtt{R}|\boldsymbol{\theta}]$ eventually becomes bounded away from 0 and the decoding delay grows to infinity.

Note that while $\boldsymbol{\theta}^{**} \in \boldsymbol{\Theta}$ by the permutation invariance of $\boldsymbol{\Theta}$, technically $\boldsymbol{\theta}^{**'}$ may not be in $\boldsymbol{\Theta}$. However, since $\mathbf{s}[0]$ can be decoded before time $\mathtt{T}$, to the decoder it is as if $\mathbf{s}[0]$ is known perfectly by time $\mathtt{T}$ and hence the decoding delay for the state sequence $\boldsymbol{\theta}^{**'}$ must be no greater than for $\boldsymbol{\theta}^{**}$. Specifically, the decoder must be able to recover $\mathbf{s}[1]$ by time $\mathtt{T}+1$. Furthermore, since $\theta^*[0]$ was chosen to maximally increase the information debt while $\theta^*[\mathtt{T}+1]$ was chosen to not increase it,

$$I(\mathbf{x}[\mathtt{T}+1];\mathbf{y}[\mathtt{T}+1]|\theta^*[\mathtt{T}+1]) - I(\mathbf{x}[0];\mathbf{y}[0]|\theta^*[0]) > \delta$$

and hence $I_d[\mathtt{T}+1,\mathtt{R}|\boldsymbol{\theta}^{**'}]$ must be at least $2\delta$.

Next, define the new channel state sequences $\boldsymbol{\theta}^{***}$, $\boldsymbol{\theta}^{***'}$ by letting $\boldsymbol{\theta}^{***} = \boldsymbol{\theta}^{**}$, and $\boldsymbol{\theta}^{***'} = \boldsymbol{\theta}^{**'}$, except that $\theta^{***}[\mathtt{T}+1] = \theta^*[1]$, $\theta^{***'}[\mathtt{T}+1] = \theta^*[1]$, and $\theta^{***}[1] = \theta^*[\mathtt{T}+1]$. Once again, $\boldsymbol{\theta}^{***'}$ may not be in $\boldsymbol{\Theta}$, but $\boldsymbol{\theta}^{***}$ is in $\boldsymbol{\Theta}$, and the decoding delay for these two is equivalent since we have already shown $\mathbf{s}[0]$ is recovered by time $\mathtt{T}-1$ and $\mathbf{s}[1]$ is receovered by time $\mathtt{T}$. Also, using similar reasoning as before, since $\theta^*[1]$ was chosen to maximally increase the information debt while $\theta^*[\mathtt{T}+2]$ was chosen to not increase it,

$$I(\mathbf{x}[\mathtt{T}+2];\mathbf{y}[\mathtt{T}+2]|\theta^*[\mathtt{T}+2]) - I(\mathbf{x}[1];\mathbf{y}[1]|\theta^*[1]) > 0$$

and hence $I_d[\mathtt{T}+2,\mathtt{R}|\boldsymbol{\theta}^{***'}]$ must still be at least $2\delta$.

We can continue this pattern to construct the sequences,

$$\theta^{*^{q'}}[i] = \begin{cases} \theta^*[i \bmod \mathtt{T}], & 0 \le i \le \mathtt{T}+q-2 \\ \theta^*[i], & \text{otherwise} \end{cases} \tag{G.8}$$

and

$$\theta^{*^{q}}[i] = \begin{cases} \theta^*[i-\mathtt{T}], & \mathtt{T} \le i \le \mathtt{T}+q-2 \\ \theta^*[i+\mathtt{T}], & 0 \le i \le q-2 \\ \theta^*[i], & \text{otherwise.} \end{cases} \tag{G.9}$$

The latter sequence is an element of $\boldsymbol{\Theta}$ and requires decoding delay at most $\mathtt{T}-1$ by assumption. Furthermore, we can show by induction that

$$I_d[\mathtt{T}+q-2,\mathtt{R}|\boldsymbol{\theta}^{*^{q'}}] > \left\lceil \frac{q-1}{2\mathtt{T}} \right\rceil \delta.$$

Choose $q$ such that $\lceil (q-1)/(2\mathtt{T}) \rceil > \mathtt{RT}/\delta$, and consider when $\mathbf{s}[q-1]$ can be decoded for the channel state sequence $\boldsymbol{\theta}^{*^{q'}}$. By induction on our previous arguments, we can show that the decoding delay for $\boldsymbol{\theta}^{q^{*'}}$ is no greater than for $\boldsymbol{\theta}^{q^*}$. Hence, the decoding delay is $\mathtt{T}-1$ and $\mathbf{s}[q-1]$ can be recovered by time $\mathtt{T}+q-2$. But the information debt at time $\mathtt{T}+q-2$ is at least $\mathtt{RT}$, and according to Lemma 6, $\mathbf{s}[q-1]$ can not be decoded at or before time $\mathtt{T}+q-2$. Therefore, it is impossible for any code to correct all patterns in $\boldsymbol{\Theta}$ with delay less than $\mathtt{T}$, and the proof is complete.

□

*Proof of Theorem 31:* By assumption the random walk defined in (9.29) will satisfy the requirements for Wald's Identity, (9.30). Taking the derivative with respect to $r$ of both sides of (9.30) and evaluating the result as $r \to 0$ yields

$$E[w[t]] = E[v]E[t]. \tag{G.10}$$

By construction, the random walk ends at time $t$ when $w[t]$ drops below $-I_d[i_s, \mathtt{R}|\boldsymbol{\theta}]$ so

$$E[v]E[t] = E[w[t]] \leq -I_d[i_s, \mathtt{R}|\boldsymbol{\theta}] \tag{G.11}$$

$$E[t] \geq \frac{-I_d[i_s, \mathtt{R}|\boldsymbol{\theta}]}{E[v]} \tag{G.12}$$

where the inequality changes direction due to division by $E[v]$ which is negative since $\mathtt{R} < E[I(x;y)]$. Substituting the definitions of $v$ (9.28) and $I_d[i, \mathtt{R}|\boldsymbol{\theta}]$ (9.9), and taking the expectation of $I_d[i_s, \mathtt{R}|\boldsymbol{\theta}]$ conditioned on the fact that $I_d[i, \mathtt{R}|\boldsymbol{\theta}]$ becomes positive at time $i = i_s$ then yields the first term in (9.31).

Next we note that $w[t-1] > -I_d[i_s, \mathtt{R}|\boldsymbol{\theta}]$ by definition and therefore

$$w[t] = w[t-1] + v\,[t] > -I_d[i_s, \mathtt{R}|\boldsymbol{\theta}] + \min v.$$

Substituting this bound into Wald's Identity and preceding as before yields the second term in (9.31). □

*Proof of Theorem 32:* We begin by taking the second derivative with respect to $r$ of Wald's Identity, (9.30), and evaluating the result as $r \to 0$:

$$E\{w[t]^2 - 2t \cdot w[t] \cdot E[v] + t^2 \cdot E[v]^2 - t \cdot \mathrm{VAR}[v]\} = 0.$$

Performing some simple algebra yields

$$\begin{array}{rcl} E[t^2] \cdot E[v]^2 \pm E[t]^2 \cdot E[v]^2 & = & -E[w[t]^2] \pm E[w[t]]^2 \\ & & +2E[t \cdot w[t]] \cdot E[v] + E[t] \cdot \mathrm{VAR}[v] \\ \mathrm{VAR}[t] \cdot E[v]^2 + (E[v] \cdot E[t])^2 & = & -\mathrm{VAR}[w[t]] - E[w[t]]^2 \\ & & +2E[t \cdot w[t]] \cdot E[v] + E[t] \cdot \mathrm{VAR}[v] \\ \mathrm{VAR}[t] \cdot E[v]^2 + 2(E[v] \cdot E[t])^2 & = & -\mathrm{VAR}[w[t]] + 2E[t \cdot w[t]] \cdot E[v] + \mathrm{VAR}[v] \cdot E[t] \end{array}$$

where the last line follows from (G.10).

Next we obtain the following sequence of inequalities

$$\frac{\mathrm{VAR}[t]}{E[t]} \leq -2E[t] - \frac{\mathrm{VAR}[w[t]]}{E[v]^2 \cdot E[t]} + \frac{2 \cdot E[t \cdot (I_d[i_s, \mathtt{R}|\boldsymbol{\theta}] + \min v)]}{E[v] \cdot E[t]} + \frac{\mathrm{VAR}[v]}{E[v]^2} \tag{G.13}$$

$$\frac{\mathrm{VAR}[t]}{E[t]} \leq -\frac{2 \cdot I_d[i_s, \mathtt{R}|\boldsymbol{\theta}]}{E[v]} - \frac{\mathrm{VAR}[w[t]]}{E[v]^2 \cdot E[t]} + \frac{2 \cdot I_d[i_s, \mathtt{R}|\boldsymbol{\theta}] + \min v}{E[v]} + \frac{\mathrm{VAR}[v]}{E[v]^2} \tag{G.14}$$

$$\leq \frac{\mathrm{VAR}[v]}{E[v]^2} + \frac{\min v}{E[v]}. \tag{G.15}$$

We obtain (G.13) by noting that $w[t]/E[v] \leq (I_d[i_s, \mathtt{R}|\boldsymbol{\theta}] + \min v)/E[v]$. Using (G.12) yields (G.14) and (G.15) follows from the obvious fact that $-\mathrm{VAR}[w[t]] \leq 0$. Substituting the appropriate quantities completes the proof. □

*Proof of Theorem 33:* Starting from (9.30) we have the following chain of inequalities:

$$\begin{aligned}
1 &= E[\exp(r \cdot \mathsf{w}[t] - \mathsf{t} \cdot \gamma(r))] && \text{(G.16)}\\
&\geq E[\exp(r \cdot \mathsf{w}[t] - \mathsf{t} \cdot \gamma(r))|\mathsf{t} > \mathtt{T}] \cdot \Pr[\mathsf{t} > \mathtt{T}] && \text{(G.17)}\\
&\geq \exp(r \cdot \mathsf{w}[t] - \mathtt{T} \cdot \gamma(r)) \cdot \Pr[\mathsf{t} > \mathtt{T}], \quad r < r^* && \text{(G.18)}\\
\Pr[\mathsf{t} > \mathtt{T}] &\leq \exp(-r \cdot \mathsf{w}[t] + \mathtt{T} \cdot \gamma(r)), \quad r < r^* && \text{(G.19)}
\end{aligned}$$

where (G.18) follows since $\gamma(r) < 0$ for $r < r^*$ by the definition of $r^*$ and the assumption that $R < E[\mathsf{I}(\mathsf{x};\mathsf{y})]$. We obtain (9.33) by noting that

$$\mathsf{w}[\mathsf{t}] \geq \mathsf{w}[\mathsf{t} - 1] + \min v \geq -\mathsf{I}_d[i_s, \mathtt{R}|\boldsymbol{\theta}] + \min v. \tag{G.20}$$

To obtain (9.34), we again start from (9.30) and use the following chain of inequalities:

$$\begin{aligned}
1 &= E[\exp(r \cdot \mathsf{w}[t] - \mathsf{t} \cdot \gamma(r))] && \text{(G.21)}\\
&\geq E[\exp(r \cdot \mathsf{w}[t] - \mathsf{t} \cdot \gamma(r))|\mathsf{t} < \mathtt{T}] \cdot \Pr[\mathsf{t} < \mathtt{T}] && \text{(G.22)}\\
&\geq \exp(r \cdot \mathsf{w}[t] - \mathtt{T} \cdot \gamma(r)) \cdot \Pr[\mathsf{t} < \mathtt{T}], \quad r > r^* && \text{(G.23)}\\
\Pr[\mathsf{t} < \mathtt{T}] &\leq \exp(-r \cdot \mathsf{w}[t] + \mathtt{T} \cdot \gamma(r)), \quad r < r^* && \text{(G.24)}
\end{aligned}$$

where (G.23) follows since $\gamma(r) > 0$ for $r > r^*$ by the definition of $r^*$ and the assumption that $R < E[\mathsf{I}(\mathsf{x};\mathsf{y})]$. We obtain (9.34) by again using (G.20).

□

Appendix H

# Distortion Exponent Derivations

## ■ H.1 Distortion Exponent For Selection Channel Coding Diversity

*Proof of Theorem 35:* The minimum expected distortion for such a scheme is computed as follows:

$$\begin{aligned} E[D] &= \min_D \Pr\left\{\max\left[I(x_1;y_1), I(x_2;y_2)\right] < R(D)\right\} \\ &\quad + \Pr\left\{\max\left[I(x1;y_1), I(x_2;y_2)\right] \geq R(D)\right\} \cdot D && \text{(H.1)} \\ &= \min_D F_{e^I}(\exp R(D))^2 + \left[1 - F_{e^I}(\exp R(D))^2\right] \cdot D && \text{(H.2)} \\ &\approx \min_D c^2 D^{\frac{-p}{\beta}} \mathrm{SNR}^{-2p} + \left(1 - c^2 D^{\frac{-p}{\beta}} \mathrm{SNR}^{-2p}\right) \cdot D && \text{(H.3)} \\ &\approx \min_D c^2 D^{\frac{-p}{\beta}} \mathrm{SNR}^{-2p} + D. && \text{(H.4)} \end{aligned}$$

Differentiating and setting equal to zero yields

$$D^* = \mathrm{SNR}^{\frac{-2p\beta}{p+\beta}} \cdot \left(\frac{\beta}{pc}\right)^{\frac{-p\beta}{p+\beta}} \tag{H.5}$$

and thus

$$E[D] \approx C_{\mathrm{SEL-CCDIV}} \mathrm{SNR}^{\frac{-2p\beta}{p+\beta}} \tag{H.6}$$

where $C_{\mathrm{SEL-CCDIV}}$ is a constant independent of SNR. □

## ■ H.2 Distortion Exponent For Multiplexed Channel Coding Diversity

*Proof of Theorem 36:* The minimum expected distortion for such a scheme is computed as follows:

$$\begin{aligned} E[D] &= \min_D \Pr\left\{\min\left[I(x_1;y_1), I(x_2;y_2)\right] < R(D)\right\} \\ &\quad + \Pr\left\{\min\left[I(x1;y_1), I(x_2;y_2)\right] \geq R(D)\right\} \cdot D && \text{(H.7)} \\ &= \min_D 2F_{e^I}(\exp[R(D)/2]) - F_{e^I}(\exp[R(D)/2])^2 + \left[1 - F_{e^I}(\exp[R(D)/2])\right]^2 \cdot D && \text{(H.8)} \\ &\approx \min_D 2cD^{\frac{-p}{4\beta}} \mathrm{SNR}^{-p} - c^2 D^{\frac{-p}{2\beta}} \mathrm{SNR}^{-2p} + \left(1 - cD^{\frac{-p}{4\beta}} \mathrm{SNR}^{-p}\right)^2 \cdot D && \text{(H.9)} \\ &\approx \min_D 2cD^{\frac{-p}{4\beta}} \mathrm{SNR}^{-p} + D && \text{(H.10)} \end{aligned}$$

Differentiating and setting equal to zero yields the optimizing distortion

$$D^* = \mathrm{SNR}^{\frac{-4p\beta}{p+4\beta}} \cdot \left(\frac{2\beta}{pc}\right)^{\frac{-4\beta}{p+4\beta}} \tag{H.11}$$

and thus

$$E[D] \approx C_{\mathrm{MPX-CCDIV}} \mathrm{SNR}^{\frac{-4p\beta}{p+4\beta}} \tag{H.12}$$

where $C_{\mathrm{MPX-CCDIV}}$ is a constant independent of SNR. □

## ■ H.3 Distortion Exponent for Optimal Channel Coding Diversity

Before proving Theorem 37 we require the following lemma characterizing the mutual information for the parallel channel in terms of probability distribution for each sub-channel.

**Lemma 10.** *Let*

$$I(\mathbf{x};\mathbf{y}) = I(x_1;y_1) + I(x_2;y_2)$$

*be the mutual information for the total channel and assume that the density and distribution for each sub-channel is given by.* (10.43) *and* (10.44) *If we define the cumulative distribution function for* $\exp I(\mathbf{x};\mathbf{y})$ *as* $F_{e^{I_0+I_1}}(t)$ *then*

$$F_{e^{I_0+I_1}}(t) \approx pc^2 \left(\frac{t}{\mathrm{SNR}^2}\right)^p \left(\ln t - \frac{1}{p}\right) \tag{H.13}$$

*in the sense that the ratio of these quantities goes to 1 as* $\mathrm{SNR} \to \infty$*.*

*Proof.* Note that for any random variable, $\boldsymbol{\theta}$ with density $f_{\boldsymbol{\theta}}(t)$, we have

$$f_{e^{\boldsymbol{\theta}}}(t) = f_{\boldsymbol{\theta}}(\ln t)/t \text{ and } f_{\boldsymbol{\theta}}(t) = f_{e^{\boldsymbol{\theta}}}(e^t) \cdot e^t. \tag{H.14}$$

Therefore we can obtain the desired result by computing the pdf, $f_{I_0+I_1}(t)$, via convolution, and applying (H.14):

$$\begin{aligned}
f_{I_0+I_1}(t) &= \int_0^t f_I(\tau) \cdot f_I(t-\tau) d\tau && \text{(H.15)}\\
&= \int_0^t e^\tau f_{e^I}(e^\tau) \cdot e^{t-\tau} f_{e^I}(e^{t-\tau}) d\tau && \text{(H.16)}\\
&\approx \int_0^t cp \frac{e^{p\tau}}{\mathrm{SNR}^p} \cdot cp \frac{e^{p(t-\tau)}}{\mathrm{SNR}^p} d\tau && \text{(H.17)}\\
&= c^2 p^2 \frac{e^{pt}}{\mathrm{SNR}^{2p}} \int_0^t d\tau && \text{(H.18)}\\
&= c^2 p^2 t \frac{e^{pt}}{\mathrm{SNR}^{2p}} && \text{(H.19)}\\
F_{e^{I_0+I_1}}(t) &= \int_{-\infty}^t f_{e^{I_0+I_1}}(\tau) d\tau && \text{(H.20)}\\
&= \int_{-\infty}^t \frac{f_{I_0+I_1}(\ln \tau)}{\tau} d\tau && \text{(H.21)}\\
&\approx \frac{c^2 p^2}{\mathrm{SNR}^{2p}} \int_1^t \tau^{p-1} \cdot \ln \tau d\tau && \text{(H.22)}\\
&= \frac{c^2 p^2}{\mathrm{SNR}^{2p}} \cdot \left(\frac{t^p \ln t}{p} - \frac{t^p}{p^2} + \frac{1}{p^2}\right) && \text{(H.23)}\\
&\approx pc^2 \left(\frac{t}{\mathrm{SNR}^2}\right)^p \left(\ln t - \frac{1}{p}\right) && \text{(H.24)}
\end{aligned}$$

where (H.17) follows from the high SNR approximation in (10.43), (H.22) follows from substituting (H.19) into (H.21) and noting that since $I(\mathbf{x};\mathbf{y})$ is positive then $f_{I_0+I_1}(\ln t)$ is non-zero only for $t > 1$, and the final line follows from noting that the last parenthesized term in (H.23) is negligible at high SNR. □

*Proof of Theorem 37:* To compute the minimum average distortion we have

$$E[D] = \min_D \Pr[I(x_1;y_1) + I(x_2;y_2) < R(D)] + \{1 - \Pr[I(x_1;y_1) + I(x_2;y_2) < R(D)]\} \cdot D \tag{H.25}$$

$$= \min_D F_{e^{I_0+I_1}}(\exp R(D)) + [1 - F_{e^{I_0+I_1}}(R(D))] \cdot D \tag{H.26}$$

$$\approx \min_D pc^2 \frac{D^{\frac{-p}{2\beta}}}{\mathrm{SNR}^{2p}} \left( -\beta \ln D - \frac{1}{p} + \frac{D^{\frac{p}{2\beta}}}{p} \right) + \left[ 1 - pc^2 \frac{D^{\frac{-p}{2\beta}}}{\mathrm{SNR}^{2p}} \left( -\beta \ln D - \frac{1}{p} + \frac{D^{\frac{p}{2\beta}}}{p} \right) \right] \cdot D \tag{H.27}$$

$$\approx \min_D pc^2 \frac{D^{\frac{-p}{2\beta}}}{\mathrm{SNR}^{2p}} \left( -\beta \ln D - \frac{1}{p} + \frac{D^{\frac{p}{2\beta}}}{p} \right) + D. \tag{H.28}$$

By noting that the parenthesized term in (H.28) is between $1/p$ and $(1 + e^{-1})/p$ when $D < \exp -(1/p\beta)$, we obtain

$$\min_D \ c^2 \frac{D^{\frac{-p}{2\beta}}}{\mathrm{SNR}^{2p}} + D \lessapprox E[D] \lessapprox \min_D \ (1 + e^{-1}) c^2 \frac{D^{\frac{-p}{2\beta}}}{\mathrm{SNR}^{2p}} + D. \tag{H.29}$$

Differentiating the lower bound and setting equal to zero yields the optimizing distortion

$$D^* = \mathrm{SNR}^{\frac{-4p\beta}{p+2\beta}} \cdot \left( \frac{c^2 p}{2\beta} \right)^{\frac{-2\beta}{p+2\beta}}. \tag{H.30}$$

Substituting (H.30) into (H.29) yields

$$C_{\mathrm{LB}} \cdot \mathrm{SNR}^{\frac{-4p\beta}{p+2\beta}} \ \lessapprox E[D] \lessapprox \ C_{\mathrm{UB}} \cdot \mathrm{SNR}^{\frac{-4p\beta}{p+2\beta}} \tag{H.31}$$

where $C_{\mathrm{LB}}$ and $C_{\mathrm{UB}}$ are terms independent of SNR. Hence we conclude that the distortion exponent is

$$\Delta_{\mathrm{OPT-CCDIV}} = (4p\beta)/(p + 2\beta). \tag{H.32}$$

□

## ■ H.4 Distortion Exponent for Source Coding Diversity

*Proof of Theorem 38:* For small $D_0$ and $D_1$, the average distortion is

$$\begin{aligned}
E[D] &= \min_{D_0,D_1} \Pr[I(x_1;x_2) < R_{\mathrm{md}}(D_0,D_1)/2]^2 \\
&\quad + 2\Pr[I(x_1;x_2) < R_{\mathrm{md}}(D_0,D_1)/2] \cdot \Pr[I(x_1;x_2) \geq R_{\mathrm{md}}(D_0,D_1)/2] \cdot D_1 \\
&\quad + \Pr[I(x_1;x_2) \geq R_{\mathrm{md}}(D_0,D_1)/2]^2 \cdot D_0 \qquad \text{(H.33)} \\
&= \min_{D_0,D_1} F_{e^I}(\exp R_{\mathrm{md}}(D_0,D_1)/2)^2 \\
&\quad + 2 \cdot F_{e^I}(\exp R_{\mathrm{md}}(D_0,D_1)/2) \cdot [1 - F_{e^I}(\exp R_{\mathrm{md}}(D_0,D_1)/2)] \cdot D_1 \\
&\quad + [1 - F_{e^I}(\exp R_{\mathrm{md}}(D_0,D_1)/2)]^2 \cdot D_0 \qquad \text{(H.34)} \\
&\approx \min_{D_0,D_1} \frac{c^2}{\mathrm{SNR}^{2p}} \exp\{p \cdot R_{\mathrm{md}}(D_0,D_1)\} \\
&\quad + 2\frac{c}{\mathrm{SNR}^p} \exp\left\{\frac{p}{2} \cdot R_{\mathrm{md}}(D_0,D_1)\right\} \cdot \left[1 - \frac{c}{\mathrm{SNR}^p} \exp\left\{\frac{p}{2} \cdot R_{\mathrm{md}}(D_0,D_1)\right\}\right] \cdot D_1 \\
&\quad + \left[1 - \frac{c}{\mathrm{SNR}^p} \exp\left\{\frac{p}{2} \cdot R_{\mathrm{md}}(D_0,D_1)\right\}\right]^2 \cdot D_0 \qquad \text{(H.35)} \\
&\approx \min_{D_0,D_1} \frac{c^2}{\mathrm{SNR}^{2p}} \exp\{p \cdot R_{\mathrm{md}}(D_0,D_1)\} + 2\frac{c}{\mathrm{SNR}^p} \exp\left\{\frac{p}{2} \cdot R_{\mathrm{md}}(D_0,D_1)\right\} \cdot D_1 + D_0. \\
&\qquad \text{(H.36)}
\end{aligned}$$

Substituting the bounds from (10.30) into (H.36) yields

$$E[D] \lessapprox \min_{D_1,D_0} \frac{c^2}{\mathrm{SNR}^{2p}} \left(\frac{1}{2D_1D_0}\right)^{\frac{p}{2\beta}} + \frac{2c}{\mathrm{SNR}^p} \left(\frac{1}{2D_1D_0}\right)^{\frac{p}{4\beta}} \cdot D_1 + D_0 \qquad \text{(H.37a)}$$

$$E[D] \gtrapprox \min_{D_1,D_0} \frac{c^2}{\mathrm{SNR}^{2p}} \left(\frac{1}{4D_1D_0}\right)^{\frac{p}{2\beta}} + \frac{2c}{\mathrm{SNR}^p} \left(\frac{1}{4D_1D_0}\right)^{\frac{p}{4\beta}} \cdot D_1 + D_0 \qquad \text{(H.37b)}$$

where (H.37b) requires $D_0 \to 0$ and (H.37a) also requires $D_1 \to 0$.

When $p \geq 4\beta$ then (H.37) increases as $D_1$ becomes small. Hence in this regime the optimal choice for $D_1$ approaches a constant bounded away from zero. If the low distortion formula for the lower bound is used, then the optimal choice for $D_1$ approaches one. Technically, however, for $D_1 \geq 1/2$ the rate required is given by (10.26b) not (10.26a), so there is no excess rate in multiple description coding [71,76] and the optimal $D_1$ for $p \geq 4\beta$ approaches 1/2 using (10.26b). In any case, regardless of whether $D_1 = 1/2$ or $D_1 = 1$ or some other intermediate value, when $p \geq 4\beta$, average distortion is minimized by choosing $D_1$ to be large. Thus for $p \geq 4\beta$, the optimal multiple description system essentially degenerates into the channel multiplexing scheme analyzed in Section 10.4.4 and achieves the same distortion exponent (although with a slightly different constant factor term).

When $p < 4\beta$, we can find the optimal value for $D_1$ by differentiating the lower bound with respect to $D_1$ and setting equal to 0 to obtain

$$D_1^* = \left(\frac{4\beta - p}{cp}\right)^{\frac{-4\beta}{4\beta-p}} \cdot \mathrm{SNR}^{\frac{-4\beta p}{4\beta+p}} \cdot (4D_0)^{-1+\frac{4\beta}{4\beta+p}}, p < 4\beta. \qquad \text{(H.38)}$$

For the case when $p < 4\beta$, substituting (H.38) into (H.37b) yields

$$E[D] \gtrapprox C \cdot D_0^{\frac{-2p}{4\beta+p}} \cdot \mathrm{SNR}^{\frac{-8p\beta}{4\beta+p}} + D_0 \text{ for } p < 4\beta \tag{H.39}$$

where $C$ is a constant independent of SNR and $D_0$. Differentiating with respect to $D_0$ and setting the result equal to zero yields the optimal value for $D_0$:

$$D_0^* = \begin{cases} C' \cdot \mathrm{SNR}^{\frac{-8\beta p}{4\beta+3p}}, & p < 4\beta \\ C'' \cdot \mathrm{SNR}^{\frac{-4\beta p}{4\beta+p}}, & p \geq 4\beta \end{cases} \tag{H.40}$$

from which we conclude

$$C_{\mathrm{LB}} \cdot \mathrm{SNR}^{-\max\left[\frac{8\beta p}{4\beta+3p}, \frac{4\beta p}{4\beta+p}\right]} \lessapprox E[D] \lessapprox C_{\mathrm{UB}} \cdot \mathrm{SNR}^{-\max\left[\frac{8\beta p}{4\beta+3p}, \frac{4\beta p}{4\beta+p}\right]} \tag{H.41}$$

where the max occurs since multiple description coding essentially degenerates into channel multiplexing with a better constant factor when $p \geq 4\beta$. □

## ■ H.5 Distortion Exponent for Source Coding Diversity with Joint Decoding

Computing the exact rates required to guarantee successful encoding in (10.65) is generally difficult, thus we focus on the high resolution limit in the following Lemma.

**Lemma 11.** *Let $\mathsf{s}$ be a source with finite variance and finite entropy power. Then in the high resolution limit, choosing*

$$R > h(\mathsf{s}) - (1/2)\log 2\pi e\sigma^2 \tag{H.42}$$

*asymptotically satisfies* (10.65) *and guarantees successful encoding.*

*Proof.* Proving the claim requires showing that

$$\lim_{D_j \to 0} I(\mathsf{s}; \hat{\mathsf{s}}_j) - \left[h(\mathsf{s}) - \frac{1}{2}\log 2\pi e\sigma^2\right] = 0 \quad \text{for } j \in \{0, 1\} \tag{H.43}$$

and

$$\lim_{D_j \to 0} I(\mathsf{s}; \hat{\mathsf{s}}_1\hat{\mathsf{s}}_2) + I(\hat{\mathsf{s}}_1; \hat{\mathsf{s}}_2) - 2\left[h(\mathsf{s}) - \frac{1}{2}\log 2\pi e\sigma^2\right] = 0. \tag{H.44}$$

The former follows from the fact that the Shannon Lower Bound is asymptotically tight [109]. In the interest of completeness, however, we define $\Delta R$ as left hand side of

(H.43) and summarize the argument showing that it goes to zero:

$$\Delta R \triangleq \lim_{D_j \to 0} I(\mathsf{s}; \hat{\mathsf{s}}_j) - \left[ h(\mathsf{s}) - \frac{1}{2} \log 2\pi e \sigma^2 \right] \tag{H.45}$$

$$= \lim_{D_j \to 0} h(\mathsf{s} + \mathsf{n}_j) - h(\mathsf{s} + \mathsf{n}_j | \mathsf{s}) - \left[ h(\mathsf{s}) - \frac{1}{2} \log 2\pi e \sigma^2 \right] \tag{H.46}$$

$$= \lim_{D_j \to 0} h(\mathsf{s} + \mathsf{n}_j) - h(\mathsf{n}_j) - \left[ h(\mathsf{s}) - \frac{1}{2} \log 2\pi e \sigma^2 \right] \tag{H.47}$$

$$= \lim_{D_j \to 0} h(\mathsf{s} + \mathsf{n}_j) - h(\mathsf{s}) \tag{H.48}$$

$$= 0. \tag{H.49}$$

Equations (H.46) and (H.47) follow from the choice of the conditional distribution $\hat{\mathsf{s}}_j = \mathsf{s} + \mathsf{n}_j$ where $\mathsf{n}_j$ is independent of $\mathsf{s}$. The key step in going from (H.48) to (H.49) is the "continuity" property of differential entropy [109, Theorem 1] which is the main tool in obtaining many high-resolution source coding results.

A similar chain of equalities establishes (H.44). Specifically, if we define the right hand side of (H.44) as $\Delta 2R$ then we obtain

$$\Delta 2R \triangleq \lim_{D_j \to 0} I(\mathsf{s}; \hat{\mathsf{s}}_1 \hat{\mathsf{s}}_2) + I(\hat{\mathsf{s}}_1; \hat{\mathsf{s}}_2) - 2 \left[ h(\mathsf{s}) - \frac{1}{2} \log 2\pi e \sigma^2 \right] \tag{H.50}$$

$$= \lim_{D_j \to 0} h(\hat{\mathsf{s}}_1 \hat{\mathsf{s}}_2) - h(\hat{\mathsf{s}}_1 \hat{\mathsf{s}}_2 | \mathsf{s}) + h(\hat{\mathsf{s}}_1) - h(\hat{\mathsf{s}}_1 | \hat{\mathsf{s}}_2) - 2 \left[ h(\mathsf{s}) - \frac{1}{2} \log 2\pi e \sigma^2 \right] \tag{H.51}$$

$$= \lim_{D_j \to 0} h(\hat{\mathsf{s}}_1) + h(\hat{\mathsf{s}}_2) - h(\hat{\mathsf{s}}_1 | \mathsf{s}) - h(\hat{\mathsf{s}}_2 | \mathsf{s}) - 2 \left[ h(\mathsf{s}) - \frac{1}{2} \log 2\pi e \sigma^2 \right] \tag{H.52}$$

$$= \lim_{D_j \to 0} 2 \cdot \Delta R \tag{H.53}$$

$$= 0 \tag{H.54}$$

where (H.53) follows by noting that (H.52) is simply twice (H.45), and hence (H.54) follows from (H.49). □

In the sequel, we require the following Lemma which states that, in the high resolution limit, the two descriptions, $\hat{\mathbf{s}}_1$ and $\hat{\mathbf{s}}_2$, only differ in half a bit per sample. This close relationship between the two descriptions enables the joint decoder to approach the performance of parallel channel coding with a single description.

**Lemma 12.** *If the rate is chosen according to* (H.42)*, specifically, if the difference between the two sides is* $\epsilon$*, then*

$$\lim_{D_j \to 0} I(\hat{\mathsf{s}}_1; \hat{\mathsf{s}}_2) - R \geq -\frac{1}{2} \log 2 - \epsilon. \tag{H.55}$$

*Proof.* We have the following chain of inequalities:

$$\begin{aligned}
\lim_{D_j \to 0} I(\hat{s}_1;\hat{s}_2) - R &= \lim_{D_j \to 0} h(s+n_1) - h(s+n_1|s+n_2) - R && \text{(H.56)}\\
&= \lim_{D_j \to 0} h(s+n_1) - h(n_1-n_2|s+n_2) - R && \text{(H.57)}\\
&\geq \lim_{D_j \to 0} h(s+n_1) - h(n_1-n_2) - R && \text{(H.58)}\\
&= \lim_{D_j \to 0} h(s+n_1) - \frac{1}{2}\log 4\pi e\sigma^2 - R && \text{(H.59)}\\
&= \lim_{D_j \to 0} h(s+n_1) - h(n_1) - \frac{1}{2}\log 2 - R && \text{(H.60)}\\
&= \lim_{D_j \to 0} I(\hat{s}_1;s) - R - \frac{1}{2}\log 2 && \text{(H.61)}\\
&= \lim_{D_j \to 0} I(\hat{s}_1;s) - \left[h(s) - \frac{1}{2}\log 2\pi e + \epsilon\right] - \frac{1}{2}\log 2 && \text{(H.62)}\\
&= \lim_{D_j \to 0} \Delta R - \frac{1}{2}\log 2 - \epsilon && \text{(H.63)}\\
&= -\frac{1}{2}\log 2 - \epsilon. && \text{(H.64)}
\end{aligned}$$

Most of the arguments follow from well-known properties of mutual information and entropy. Equation (H.64) follows from Lemma 11.

□

*Proof of Theorem 40:* If we choose $\sigma^2$ as in (10.70), the expected distortion is at most the distortion when both descriptions are successfully decoded times the probability that both descriptions are not decoded. Hence, applying Theorem 39 yields

$$E[D] \leq \frac{\sigma^2}{2} \cdot \Pr[\mathcal{E}] + \Pr[\mathcal{E}^c] \tag{H.65}$$

where $\mathcal{E}$ denotes the event that both descriptions can be decoded as defined in (10.66) and $\mathcal{E}^c$ is the complement of $\mathcal{E}$. Note that since $\Pr[\mathcal{E}] \leq 1$, the first term on the right hand side of (H.65) is proportional to $\mathrm{SNR}^{-\Delta_{\mathrm{OPT-CCDIV}}}$ by construction due to our choice of $\sigma^2$ in (10.70). Therefore, to prove the Theorem, we need to bound the second term, $\Pr[\mathcal{E}^c]$.

If we let $\mathcal{E}[i,j]$ (with $i,j \in \{1,2\}$) denote the event that the first max operation in $\mathcal{E}$ returns the $i$th argument while the second max operation in $\mathcal{E}$ returns the $j$th argument, then we can express the second term in (H.65) as

$$\begin{aligned}
\Pr[\mathcal{E}^c] = \Pr[\mathcal{E}^c \cap \mathcal{E}[1,1]|\mathcal{E}[1,1]]\Pr[\mathcal{E}[1,1]] + \Pr[\mathcal{E}^c \cap \mathcal{E}[1,2]|\mathcal{E}[1,2]]\Pr[\mathcal{E}[1,2]]&\\
+ \Pr[\mathcal{E}^c \cap \mathcal{E}[2,1]|\mathcal{E}[2,1]]\Pr[\mathcal{E}[2,1]] + \Pr[\mathcal{E}^c \cap \mathcal{E}[2,2]|\mathcal{E}[2,2]]\Pr[\mathcal{E}[2,2]].& \quad \text{(H.66)}
\end{aligned}$$

To prove the theorem, it is sufficient to show that for every $\epsilon > 0$, there exists a constant $c_{i,j}$ such that

$$\Pr[\mathcal{E}^c|\mathcal{E}[i,j]]\Pr[\mathcal{E}[i,j]] \leq c_{i,j} \cdot \mathrm{SNR}^{\epsilon - \Delta_{\mathrm{OPT-CCDIV}}}$$

for large enough SNR.

Conditioned on $\mathcal{E}[1,1]$, both $I(x_1;y_1) > R/\beta$ and $I(x_2;y_2) > R/\beta$, so both channels are good enough to decode each description separately. Thus $\Pr[\mathcal{E}^c|\mathcal{E}[1,1]] = 0$, and therefore

$\Pr[\mathcal{E}^c|\mathcal{E}[1,1]]\Pr[\mathcal{E}[1,1]] = 0$ as well. This takes care of the first term in (H.66).

Next we consider the second term of (H.66). Conditioned on $\mathcal{E}[1,2]$, only $\mathsf{I}(\mathsf{x}_1;\mathsf{y}_1) > R/\beta$ while $\mathsf{I}(\mathsf{x}_2;\mathsf{y}_2) < R/\beta$ and only description 1 can be decoded separately. Description 2 can be decoded jointly provided that $\mathsf{I}(\mathsf{x}_2;\mathsf{y}_2) \geq R/\beta - I(\hat{\mathsf{s}}_1;\hat{\mathsf{s}}_2)/\beta$. By applying Lemma 12, this condition becomes $\mathsf{I}(\mathsf{x}_2;\mathsf{y}_2) > (\log 2)/(2\beta)$ in the high-resolution limit, therefore

$$\begin{aligned}
\Pr[\mathcal{E}^c|\mathcal{E}[1,2]]\Pr[\mathcal{E}[1,2]] &\approx \Pr\left[\mathsf{I}(\mathsf{x}_2;\mathsf{y}_2) \leq \frac{\log 2}{2\beta}\right] \cdot \Pr[\mathcal{E}[1,2]] && \text{(H.67)}\\
&\approx c \cdot \left(\frac{2^{\frac{1}{2\beta}}}{\mathrm{SNR}}\right)^p \cdot \Pr[\mathcal{E}[1,2]] && \text{(H.68)}\\
&\leq c \cdot \left(\frac{2^{\frac{1}{2\beta}}}{\mathrm{SNR}}\right)^p \cdot \Pr[\mathsf{I}(\mathsf{x}_2;\mathsf{y}_2) < R/\beta] && \text{(H.69)}\\
&\approx c \cdot \left(\frac{2^{\frac{1}{2\beta}}}{\mathrm{SNR}}\right)^p \cdot c \cdot \left(\frac{\exp\frac{h(\mathsf{s})}{\beta}}{\sigma^{1/\beta}\mathrm{SNR}}\right)^p && \text{(H.70)}\\
&= \mathrm{SNR}^{-2p} \cdot \mathrm{SNR}^{\frac{2p}{p+2\beta}} \cdot c^2 \left(2^{\frac{1}{2\beta}} \exp\frac{h(\mathsf{s})}{\beta}\right)^p && \text{(H.71)}\\
&= \mathrm{SNR}^{\frac{-4p\beta}{p+2\beta}} \cdot c^2 \left(2^{\frac{1}{2\beta}} \exp\frac{h(\mathsf{s})}{\beta}\right)^p && \text{(H.72)}
\end{aligned}$$

where in going from (H.69) to (H.70) we replaced $R$ with $h(\mathsf{s}) - (1/2)\log 2\pi e\sigma^2$ and recalled that we assumed $\exp[2h(\mathsf{s})] = 2\pi e$ just after (10.23).

Thus, for some constant $C_{\mathrm{SCDIV-JD}}$, and every $\epsilon > 0$, there exists an SNR large enough such that

$$\Pr[\mathcal{E}^c|\mathcal{E}[1,2]]\Pr[\mathcal{E}[1,2]] \leq \mathrm{SNR}^{\epsilon - \frac{4p\beta}{p+2\beta}} \cdot C_{\mathrm{SCDIV-JD}} \tag{H.73}$$

and

$$\Pr[\mathcal{E}^c|\mathcal{E}[2,1]]\Pr[\mathcal{E}[2,1]] \leq \mathrm{SNR}^{\epsilon - \frac{4p\beta}{p+2\beta}} \cdot C_{\mathrm{SCDIV-JD}}. \tag{H.74}$$

A similar analysis works for the third term of (H.66).

Finally, we consider the last term in (H.66). Conditioned on $\mathcal{E}[2,2]$, both $\mathsf{I}(\mathsf{x}_1;\mathsf{y}_1) < R/\beta$ and $\mathsf{I}(\mathsf{x}_2;\mathsf{y}_2) < R/\beta$, so neither channels is good enough for separate decoding. Successful joint decoding requires

$$\mathsf{I}(\mathsf{x}_1;\mathsf{y}_1) + \mathsf{I}(\mathsf{x}_2;\mathsf{y}_2) > \left[2R - I(\hat{\mathsf{s}}_1;\hat{\mathsf{s}}_2)\right]/\beta. \tag{H.75}$$

and therefore

$$\begin{aligned}\Pr[\mathcal{E}^c \cap \mathcal{E}[2,2]|\mathcal{E}[2,2]] &= \Pr\left[\mathsf{I}(\mathsf{x}_1;\mathsf{y}_1) + \mathsf{I}(\mathsf{x}_2;\mathsf{y}_2) \le 2R/\beta - I(\hat{\mathsf{s}}_1;\hat{\mathsf{s}}_2)/\beta\right] && \text{(H.76)}\\ &\lessapprox \Pr\left[\mathsf{I}(\mathsf{x}_1;\mathsf{y}_1) + \mathsf{I}(\mathsf{x}_2;\mathsf{y}_2) \le R/\beta - \frac{\log 2}{2\beta}\right] && \text{(H.77)}\\ &\approx pc^2 \left(\frac{2^{\frac{1}{2\beta}} \exp \frac{h(\mathsf{s})}{2\beta}}{\sigma^{1/\beta}\mathrm{SNR}^2}\right)^p \cdot \left(\frac{h(\mathsf{s})}{2\beta} - \frac{\log 2\pi e \sigma^2}{2\beta} + \frac{\log 2}{2\beta} - \frac{1}{p}\right) && \text{(H.78)}\\ &= \sigma^{-p/\beta} \cdot \mathrm{SNR}^{-2p} \cdot 2^{\frac{1}{2\beta}} \cdot \exp \frac{h(\mathsf{s})}{2\beta} \cdot \left(\frac{h(\mathsf{s})}{2\beta} - \frac{\log 2\pi e \sigma^2}{2\beta} + \frac{\log 2}{2\beta} - \frac{1}{p}\right) && \text{(H.79)}\\ &\approx \mathrm{SNR}^{\frac{-2p^2}{p+2\beta} - 2p^2} \cdot C'_{\mathrm{SCDIV-JD}} \cdot \mathrm{SNR}^{\epsilon} && \text{(H.80)}\\ &= \mathrm{SNR}^{\epsilon - \frac{4p\beta}{p+2\beta}} \cdot C'_{\mathrm{SCDIV-JD}} && \text{(H.81)}\end{aligned}$$

where (H.77) follows since Lemma 12 implies

$$2R - I(\hat{\mathsf{s}}_1;\hat{\mathsf{s}}_2) \le R - \frac{1}{2}\log 2 + \epsilon, \qquad \text{(H.82)}$$

$\epsilon$ is a quantity which can be made arbitrarily small, and $C'_{\mathrm{SCDIV-JD}}$ is some constant independent of SNR.

The above results combined with $\Delta_{\mathrm{OPT-CCDIV}} = 4p\beta/(p+2\beta)$ proves the desired result. □

# Bibliography

[1] M. Adams and V. Guillemin. *Measure Theory and Probability*. Birkäuser, 1996. *Cited on page* 199

[2] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, 48(8):2201–2214, August 2002. *Cited on page* 55, 66

[3] R. F. Ahlswede. The Rate-Distortion Region for Multiple Descriptions without Excess Rate. *IEEE Trans. Inform. Theory*, 31(6):721–726, November 1985. *Cited on page* 158, 161

[4] M. Alasti, K. Sayrafian-Pour, A. Ephremides, and N. Farvardin. Multiple Description Coding in Networks with Congestion Problem. *IEEE Trans. Inform. Theory*, 47(3):891–902, March 2001. *Cited on page* 158, 162, 192

[5] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan. Priority encoding transmission. *IEEE Transactions on Information Theory*, 42(6):1737–1744, Nov 1996. *Cited on page* 22

[6] E. Altman, C. Barakat, and V. Ramos. Queueing analysis of simple fec schemes for ip telephony. In *INFOCOM 2001*, volume 2, pages 796–804, 2001. *Cited on page* 22

[7] E. Altman and A. Jean-Marie. Loss probabilities for messages with redundant packets feeding a finite buffer. *Journal of Selected Areas in Communications*, 16(5):778–787, June 1998. *Cited on page* 22

[8] J. Alvarez and B. Hajek. On the use of packet classes in communication networks to enhance congestion pricing based on marks. *IEEE Transactions on Automatic Control*, 47(6):1020–1026, June 2002. *Cited on page* 22

[9] J. G. Apostolopoulos. Reliable video compression over lossy packet networks using multiple state encoding and path diversity. In *Proc. SPIE Visual Communications and Image Processing (VCIP)*, volume 4310, San Jose, CA, January 2001. *Cited on page* 158, 161

[10] J. G. Apostolopoulos, W.-T. Tan, S. J. Wee, and G. W. Wornell. Modeling Path Diversity for Multiple Description Video Communications. Proc. International Conference on Acoustics, Speech, and Signal Processing, May 2002. *Cited on page* 158, 161

[11] J. G. Apostolopoulos, W. tian Tan, and S. J. Wee. Performance of a multiple description streaming media content delivery network. In *Proc. International Conference on Image Processing*, volume 2, pages 189–192, September 2002. *Cited on page* 192

[12] J. G. Apostolopoulos and S. J. Wee. Unbalanced multiple description video communication using path diversity. In *Proc. IEEE International Conf. on Image Proc. (ICIP)*, volume 1, pages 966–969, Thessaloniki, Greece, October 2001. *Cited on page* 158

[13] M. Arai, A. Chiba, and K. Iwasaki. Measrurement and modeling of burst packet losses in internet end-to-end communications. In *Proceedings of the 1999 Pacific Rim International Symposium on Dependable Computing*, pages 260–267, 1999. *Cited on page* 102

[14] N. At and Y. Altunbasak. Multiple Description Coding for Wireless Channels with Multiple Antennas. In *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, volume 3, pages 2040–2044, San Antonio, TX, November 2001. *Cited on page* 158

[15] H. Balakrishnan, S. Seshan, and R. H. Katz. Improving reliable transport and handoff performance in cellular wireless networks. *ACM Wireless Networks*, December 1995. *Cited on page* 22

[16] R. J. Barron, B. Chen, and G. W. Wornell. The duality between information embedding and source coding with side information and some applications. *IEEE Transactions on Information Theory*, 49(5):1159–1180, May 2003. *Cited on page* 84

[17] J. Barros, J. Hagenauer, and N. Görtz. Turbo Cross Decoding of Multiple Descriptions. In *Proc. International Conference on Communications*, volume 3, pages 1398–1402, New York, NY, 20 April - 2 May 2002. *Cited on page* 158

[18] T. Berger. Private Communication. *Cited on page* 34, 35

[19] T. Berger. *Rate Distortion Theory: A Mathematical Basis For Data Compression.* Prentice-Hall, Englewood Cliffs, NJ, 1971. *Cited on page* 31, 35, 43

[20] T. Berger and Z. Zhang. Minimum Breakdown Degradation in Binary Source Encoding. *IEEE Trans. Inform. Theory*, 29(6):807–814, November 1983. *Cited on page* 161

[21] T. Berger, Z. Zhang, and H. Viswanathan. The ceo problem [multiterminal source coding]. *IEEE Transactions on Information Theory*, 42(3):887–902, May 1996. *Cited on page* 50

[22] R. A. Berry. *Power and Delay Trade-Offs in Fading Channels.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 2000. *Cited on page* 22

[23] E. Biglieri, J. Proakis, and S. Shamai. Fading channels: information-theoretic and communications aspects. *IEEE Transactions on Information Theory*, 44(6):2619–2692, October 1998. *Cited on page* 22, 96, 97, 105, 140, 151

[24] N. Blachman. The convolution inequality for entropy powers. *IEEE Transactions on Information Theory*, 11(2):267–271, April 1965. *Cited on page* 205

[25] D. Blackwell, L. Breiman, and A. J. Thomasian. The capacity of a class of channels. *Ann. Math. Stat.*, 30:1229–1241, December 1959. *Cited on page* 24, 97, 138

[26] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. IETF RFC2475, 1998. Available online at http://www.ietf.org/rfc/rfc2475.txt. *Cited on page* 22

[27] J. Bolot. Characterizing end-to-end packet delay and loss in the internet. *Journal of High Speed Networks*, pages 289–298, September 1993. *Cited on page* 102

[28] J. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive fec-based error control for internet telephony. In *Proc. of Infocom'99*, March 1999. *Cited on page* 22

[29] M. S. Borella, D. Swider, S. Uludag, and G. B. Brewster. Internet packet loss: measurement and implications for end-to-end qos. In *Proceedings of the 1998 ICPP Workshops on Architectural and OS Support for Multimedia Applications/Flexible Communications Systems/Wireless Networks and Mobile Computing*, pages 3–12, 1998. *Cited on page* 102

[30] V. S. Borkar, S. K. Mitter, and S. Tatikonda. Optimal sequential vector quantization of Markov sources. *SIAM J. Control Optim.*, 40(1):135–148, 2001. *Cited on page* 84, 190

[31] J. Byers, M. Luby, and M. Mitzenmacher. A digital fountain approach to asynchronous reliable multicast. *IEEE Transactions on Selected Areas In Communication*, 20(8):1528–1540, October 2002. *Cited on page* 25, 98, 138

[32] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. In *Proc. ACM SIGCOMM 1998*, pages 56–67, Vancouver Canada, September 1998. *Cited on page* 22, 25, 138

[33] J. W. Byers, M. Luby, and M. Mitzenmacher. Accessing multiple mirror sites in parallel: using tornado codes to speed up downloads. *Proc. INFOCOM*, pages 275–283, March 1999. *Cited on page* 25, 138, 140

[34] G. Caire and S. Shamai (Shitz). On Achievable Throughput of a Multi-Antenna Gaussian Broadcast Channel. *IEEE Trans. Inform. Theory*, 49(7):1691–1706, July 2003. *Cited on page* 158

[35] G. Caire, G. Taricco, and E. Biglieri. Optimum power control over fading channels. *IEEE Transactions on Information Theory*, 45(5):1468–1489, July 1999. *Cited on page* 22

[36] B. Chen and G. Wornell. Efficient channel coding for analog sources using chaotic systems. In *Global Telecommunications Conference*, volume 1, pages 131–135, 1996. *Cited on page* 158

[37] B. Chen and G. W. Wornell. Analog Error-Correcting Codes Based on Chaotic Dynamical Systems. *IEEE Trans. Commun.*, 46(7):881–890, July 1998. *Cited on page* 158

[38] S.-Y. Chung. *On the Construction of Some Capacity Approaching Coding Schemes.* PhD thesis, Massachusetts Institute of Technology, 2000. *Cited on page* 158

[39] T. P. Coleman and M. Medard. A distributed scheme for achieving energy-delay tradeoffs with multiple service classes over a dynamically varying network. *IEEE Transactions on Selected Areas In Communication*, 22(5):929–941, June 2004. *Cited on page* 22

[40] J. Conway and N. Sloane. Voronoi regions of lattices, second moments of polytopes, and quantization. *IEEE Transactions on Information Theory*, 28(2):211–226, March 1982. *Cited on page* 55, 68

[41] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices, and Groups.* Springer-Verlag, New York, 1988. *Cited on page* 55, 62, 70

[42] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT Electrical and Computer Science Series. MIT Press, 1992. ISBN 0-262-03141-8. *Cited on page* 148

[43] T. M. Cover. Comments on broadcast channels. *IEEE Transactions on Information Theory*, 44(6):2524–2530, October 1998. *Cited on page* 24, 138, 139, 158

[44] T. M. Cover and M. Chiang. Duality between channel capacity and rate distortion with two-sided state information. *IEEE Transactions on Information Theory*, 48(6):1629–1638, June 2002. *Cited on page* 30, 35, 198

[45] T. M. Cover and A. A. E. Gamal. Capacity theorems for the relay channel. *IEEE Transactions on Information Theory*, pages 572–584, September 1979. *Cited on page* 97

[46] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., New York, 1991. *Cited on page* 143, 158, 168, 192, 197, 199, 203, 205

[47] H. Coward, R. Knopp, and S. Servetto. On the Performance of a Natural Class of Joint Source/Channel Codes based upon Multiple Descriptions. *IEEE Trans. Inform. Theory*, 2002. Submitted for publication. Available at `http://people.ece.cornell.edu/servetto/publications/papers/20010820/`. *Cited on page* 158, 162

[48] H. Coward, R. Knopp, and S. D. Servetto. On the Performance of Multiple Description Codes over Bit Error Channels. In *Proc. International Symposium on Information Theory*, Washington, DC, July 2001. *Cited on page* 158, 162

[49] I. Csiszár. On an extremum problem of information theory. *Stud. Sci. Math. Hung.*, pages 57–70, 1974. *Cited on page* 199

[50] I. Csiszár and J. Körner. *Information Theory: Coding Theorems For Discrete Memoryless Systems.* Academic Press, 1981. *Cited on page* 30, 31, 34, 35, 97

[51] A. Dembo, T. M. Cover, and J. A. Thomas. Information theoretic inequalities. *IEEE Transactions on Information Theory*, 37(6):1501–1518, November 1991. *Cited on page* 205

[52] B. L. Douglas, S. D. Kent, and H. Lee. Parameter estimation and the importance of phase in acoustic microscopy. In *Proceedings of the IEEE Ultrasonics Symposium*, volume 2, pages 715–718, October 1992. *Cited on page* 37

[53] S. C. Draper, B. J. Frey, and F. R. Kschischang. Efficient variable length channel coding for unknown dmcs. In *Proc. International Symposium on Information Theory*, Chicago, IL, 2004. *Cited on page* 98

[54] I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. In *Proc. Foundations of Computer Science*, pages 475–484, October 1999. *Cited on page* 66

[55] N. Elia and S. K. Mitter. Stabilization of linear systems with limited information. *IEEE Transactions on Automatic Control*, 46(9):1384–1400, September 2001. *Cited on page* 84

[56] A. Ephremides and B. Hajek. Information theory and communication networks: an unconsummated union. *IEEE Transactions on Information Theory*, 44(6):2416–2434, October 1998. *Cited on page* 192

[57] W. H. R. Equitz and T. M. Cover. Successive refinement of information. *IEEE Trans. Inform. Theory*, 37(2):269–275, March 1991. *Cited on page* 158, 161

[58] M. V. Eyuboglu and G. D. Forney, Jr. Lattice and trellis quantization with lattice- and trellis-bounded codebooks high-rate theory for memoryless sources. *IEEE Transactions on Information Theory*, 39(1):46–59, January 1993. *Cited on page* 55, 68, 69

[59] M. Feder and N. Shulman. Source broadcasting with unknown amount of receiver side information. In *Proc. Information Theory Workshop*, pages 127–130, 2002. *Cited on page* 98

[60] H. Feng and M. Effros. Improved bounds for the rate loss of multiresolution source codes. *IEEE Transactions on Information Theory*, 49(4):809–821, April 2003. *Cited on page* 47

[61] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. *Transactions on Networking*, 7(4):458–472, August 1999. *Cited on page* 22

[62] G. D. Forney, Jr. Burst-correcting codes for the classic bursty channel. *IEEE Transactions on Communications Technology*, COM-19(5):772–781, October 1971. *Cited on page* 23, 122, 125

[63] G. D. Forney, Jr. Coset codes. I. introduction and geometrical classification. *IEEE Transactions on Information Theory*, 34(5):1123–1151, September 1988. *Cited on page* 55, 62, 68

[64] G. D. Forney, Jr. Multidimensional constellations. II. Voronoi constellations. *Selected Areas in Communications*, 7(6):941–958, August 1989. *Cited on page* 55, 68

[65] G. D. Forney, Jr. Geometrically uniform codes. *IEEE Transactions on Information Theory*, 37(5):1241–1260, September 1991. *Cited on page* 62

[66] G. D. Forney, Jr. Codes on graphs: normal realizations. *IEEE Transactions on Information Theory*, 47(2):520–548, Feb 2001. *Cited on page* 76, 77, 78

[67] G. Franceschetti, S. Merolla, and M. Tesauro. Phase quantized sar signal processing: theory and experiments. *IEEE Transactions on Aerospace and Electronic Systems*, 35(1):201–204, January 1999. *Cited on page* 37

[68] Y. Frank-Dayan and R. Zamir. Dithered lattice-based quantizers for multiple descriptions. *IEEE Trans. Inform. Theory*, 48(1):192–204, January 2002. *Cited on page* 158, 161

[69] R. G. Gallager. *Information Theory and Reliable Communication.* John Wiley and Sons, Inc., 1968. *Cited on page* 23

[70] R. G. Gallager. *Discrete Stochastic Processes.* Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1996. *Cited on page* 151

[71] A. A. E. Gamal and T. M. Cover. Achievable rates for multiple descriptions. *IEEE Transactions on Information Theory*, 28(6):851–857, November 1982. *Cited on page* 158, 159, 161, 168, 184, 185, 231

[72] A. Gersho. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, 25(4):373–380, July 1979. *Cited on page* 69

[73] E. N. Gilbert. Capacity of a burst-noise channel. *Bell Systems Technical Journal*, 39:1253–1265, September 1960. *Cited on page* 102

[74] H. Gish and J. Pierce. Asymptotically efficient quantizing. *IEEE Transactions on Information Theory*, 14(5):676–683, September 1968. *Cited on page* 69

[75] V. K. Goyal. Multiple Description Coding: Compression Meets the Network. *IEEE Signal Proc. Mag.*, 18(5):74–93, September 2001. *Cited on page* 158, 161, 192

[76] V. K. Goyal and J. Kovacevic. Generalized Multiple Description Coding with Correlating Transforms. *IEEE Trans. Inform. Theory*, 47(6):2199–2224, September 2001. *Cited on page* 158, 161, 173, 231

[77] R. Gray. A new class of lower bounds to information rates of stationary sources via conditional rate-distortion functions. *IEEE Transactions on Information Theory*, 19(4):480–489, July 1973. *Cited on page* 34, 35

[78] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383, October 1998. *Cited on page* 55

[79] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383, October 1998. *Cited on page* 74

[80] R. M. Gray and T. G. Stockham, Jr. Dithered quantizers. *IEEE Transactions on Information Theory*, 39(3):805–812, May 1993. *Cited on page* 69

[81] S. Hanly and D. Tse. Multiaccess fading channels II: Delay-limited capacities. *IEEE Transactions on Information Theory*, 44(7):2816–2831, November 1998. *Cited on page* 97, 105

[82] B. Hassibi and H. Vikalo. On the expected complexity of integer least-squares problems. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1497–1500, 2002. *Cited on page* 66

[83] M. Hayes, J. Lim, and A. Oppenheim. Phase-only signal reconstruction. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 437–440, 1980. *Cited on page* 37

[84] A. Hung and T. Meng. Multidimensional rotations for robust quantization of image data. *IEEE Transactions on Image Processing*, 7(1):1–12, Jan 1998. *Cited on page* 37

[85] N. Jayant, J. Johnston, and R. Safranek. Signal compression based on models of human perception. *Proceedings of the IEEE*, 81(10):1385–1422, October 1993. *Cited on page* 20, 30, 54

[86] D. G. Jeong and J. D. Gibson. Uniform and piecewise uniform lattice vector quantization for memoryless gaussian and laplacian sources. *IEEE Transactions on Information Theory*, 39(3):786–804, May 1993. *Cited on page* 55

[87] R. Johannesson and K. S. Zigangirov. *Fundamentals of Convolutional Coding.* IEEE Press, New York, NY, 1999. *Cited on page* 147

[88] R. Johari and D. K. H. Tan. End-to-end congestion control for the internet: delays and stability. *IEEE/ACM Trans. Networking*, 9(6):818–832, December 2001. *Cited on page* 22

[89] N. Kamaci, Y. Altunbasak, and R. M. Mersereau. Multiple Description Coding with Multiple Transmit and Receive Antennas for Wireless Channels: The Case of Digital Modulation. In *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, volume 6, pages 3272–3276, San Antonio, TX, November 2001. *Cited on page* 158

[90] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwidth-delay product networks. In *Proc. SIGCOMM*, 2002. *Cited on page* 22

[91] C.-S. Kim and S.-U. Lee. Multiple description coding of motion fields for robust video transmission. *IEEE Transactions On Circuits and Systems for Video Technology*, 11(9):999–1010, September 2001. *Cited on page* 158, 161

[92] A. Kirac and P. P. Vaidyanathan. Results on lattice vector quantization with dithering. *IEEE Transactions On Circuits and Systems II: Analog and Digital Signal Processing*, 43(12):811–826, December 1996. *Cited on page* 69

[93] R. Knopp and P. Humblet. On coding for block fading channels. *IEEE Transactions on Information Theory*, 46(1):189–205, Jan 2000. *Cited on page* 22, 96, 102, 140

[94] S. Kullback. *Information Theory and Statistics.* Dover, New York, 1968. *Cited on page* 206

[95] J. N. Laneman, E. Martinian, G. W. Wornell, and J. G. Apostolopoulos. Source-channel diversity approaches for multimedia communication. In *Proc. International Symposium on Information Theory*, Chicago, IL, 2004. *Cited on page* 8

[96] J. N. Laneman, E. Martinian, G. W. Wornell, J. G. Apostolopoulos, and S. J. Wee. Comparing Application- and Physical-Layer Approaches to Diversity on Wireless Channels. In *Proc. IEEE International Communications Conference (ICC)*, May 2003. *Cited on page* 8, 164, 180

[97] J. N. Laneman and G. W. Wornell. Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks. *IEEE Transactions on Information Theory*, 49(10):2415–2425, October 2003. *Cited on page* 22

[98] J. N. Laneman and G. W. Wornell. Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks. *IEEE Trans. Inform. Theory*, 49(10):2415–2425, October 2003. *Cited on page* 176, 180

[99] A. Lapidoth. The performance of convolutional codes on the block erasure channel using various finite interleaving techniques. *IEEE Transactions on Information Theory*, 40(5):1459–1473, September 1994. *Cited on page* 22

[100] A. Lapidoth. On the role of mismatch in rate distortion theory. *IEEE Trans. Inform. Theory*, 43(1):38–47, January 1997. *Cited on page* 168

[101] A. Lapidoth and I. Telatar. The compound channel capacity of a class of finite-state channels. *IEEE Transactions on Information Theory*, 44(3):973–983, May 1998. *Cited on page* 24, 138

[102] A. Lapidoth and J. Ziv. On the universality of the lz-based decoding algorithm. *IEEE Transactions on Information Theory*, 44(5):1746–1755, Sep 1998. *Cited on page* 139

[103] A. Lapidoth and J. Ziv. On the universality of the LZ-based decoding algorithm. *IEEE Transactions on Information Theory*, 44(5):1746–1755, Sep 1998. *Cited on page* 139

[104] L. Lastras and T. Berger. All sources are nearly successively refinable. *IEEE Transactions on Information Theory*, 47(3):918–926, March 2001. *Cited on page* 47

[105] L. Li and A. J. Goldsmith. Capacity and Optimal Resource Allocation for Fading Broadcast Channels – Part I: Ergodic Capacity. *IEEE Trans. Inform. Theory*, 47(3):1083–1102, March 2001. *Cited on page* 158

[106] L. Li and A. J. Goldsmith. Capacity and Optimal Resource Allocation for Fading Broadcast Channels – Part II: Outage Capacity. *IEEE Trans. Inform. Theory*, 47(3):1103–1127, March 2001. *Cited on page* 158

[107] Z.-P. Liang and P. C. Lauterbur. *Principles of Magnetic Resonance Imaging*. IEEE Press, 1999. *Cited on page* 37

[108] S. Lin and J. D. J. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice Hall, Englewood Cliffs, 1983. *Cited on page* 128

[109] T. Linder and R. Zamir. On the asymptotic tightness of the Shannon lower bound. *IEEE Transactions on Information Theory*, 40(6):2026–2031, November 1994. *Cited on page* 41, 167, 198, 202, 203, 204, 205, 232, 233

[110] T. Linder, R. Zamir, and K. Zeger. On source coding with side-information-dependent distortion measures. *IEEE Transactions on Information Theory*, 46(7):2697–2704, November 2000. *Cited on page* 31, 42, 44

[111] H. Loeliger. Averaging bounds for lattices and linear codes. *IEEE Transactions on Information Theory*, 43(6):1767–1773, November 1997. *Cited on page* 55

[112] T. D. Lookabaugh and R. M. Gray. High-resolution quantization theory and the vector quantizer advantage. *IEEE Transactions on Information Theory*, 35(5):1020–1033, September 1989. *Cited on page* 55, 68

[113] S. H. Low, F. Paganini, and J. C. Doyle. Internet congestion control. *IEEE Control Systems Magazine*, 22(1):28–43, February 2002. *Cited on page* 22

[114] M. Luby. LT codes. *Proc. Foundations of Computer Science*, November 2002. *Cited on page* 25, 138, 140

[115] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. S. and. Efficient erasure correcting codes. *IEEE Transactions on Information Theory*, 47(2):569–584, February 2001. *Cited on page* 98

[116] J. K. MacKie-Mason and H. R. Varian. Pricing the Internet. In *Public Access to the Internet, JFK School of Government, May 26–27, 1993*, page 37, 1993. *Cited on page* 22

[117] M. Marcellin and T. Fischer. Trellis coded quantization of memoryless and gauss-markov sources. *IEEE Transactions on Communications*, 38(1):82–93, January 1990. *Cited on page* 55, 74

[118] E. Martinian. Waterfilling gains O(1/SNR) at high SNR. Unpublished notes available from `http://www.csua.berkeley.edu/~emin/research/wfill.pdf`. *Cited on page* 176

[119] E. Martinian and C.-E. Sundberg. Burst erasure correction codes with low decoding delay. accepted for publication in *IEEE Transactions on Information Theory.* *Cited on page* 23, 102, 106, 108, 119, 144

[120] E. Martinian and C.-E. Sundberg. Low delay burst erasure correction codes. In *Proc. International Conference on Communications*, volume 3, pages 1736–1740, 2002. *Cited on page* 144

[121] E. Martinian and C. E. W. Sundberg. Decreasing distortion using low delay codes for bursty packet loss channels. *IEEE Transactions on Multimedia*, 5(3):285–292, September 2003. *Cited on page* 102

[122] E. Martinian and G. W. Wornell. Universal codes for minimizing per-user delay on streaming broadcast channels. In *Proc. Allerton Conf. Communications, Control, and Computing*, Monticello, IL, October 2003. *Cited on page* 8

[123] E. Martinian, G. W. Wornell, and R. Zamir. Encoder side information is useful in source coding. In *Proc. International Symposium on Information Theory*, Chicago, IL, 2004. *Cited on page* 8

[124] E. Martinian, G. W. Wornell, and R. Zamir. Source coding with distortion side information at the encoder. In *Proc. Data Compression Conference*, pages 172–181, March 2004. *Cited on page* 8, 57

[125] E. Martinian and J. S. Yedidia. Iterative quantization using codes on graphs. In *Proc. Allerton Conf. Communications, Control, and Computing*, Monticello, IL, October 2003. *Cited on page* 8, 65, 68

[126] R. McEliece and W. S. and. Channels with block interference. *IEEE Transactions on Information Theory*, 30(1):44–53, Jan 1984. *Cited on page* 96, 140

[127] U. Mittal and N. Phamdo. Hybrid Digital-Analog (HDA) Joint Source-Channel Codes for Broadcasting and Robust Communications. *IEEE Trans. Inform. Theory*, 48(5):1082–1102, May 2002. *Cited on page* 158

[128] S. K. Mitter. Control with limited information: the role of systems theory and information theory. ISIT 2000 Plenary Talk reprinted in IEEE Information Theory Society Newsletter, Decemeber 2000. pp. 1–23. *Cited on page* 190

[129] S. K. Mitter. System science: The convergence of communication, computation and control. Plenary Lecture at IEEE Conference on Control Applications, September 2002. *Cited on page* 190

[130] N. Moayeri and D. L. Neuhoff. Theory of lattice-based fine-coarse vector quantization. *IEEE Transactions on Information Theory*, 37(4):1072–1084, July 1991. *Cited on page* 55

[131] D. Mukherjee and S. K. Mitra. Successive refinement lattice vector quantization. *IEEE Transactions on Image Processing*, 11(12):1337–1348, December 2002. *Cited on page* 55

[132] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein. Future directions in control in an information-rich world. *IEEE Control Systems Magazine*, 23(2):20–33, April 2003. *Cited on page* 84, 190

[133] K. Nahrstedt and S. Servetto. Video streaming over the public internet: multiple description codes and adaptive transport protocols. In *Proc. International Conference on Image Processing*, volume 3, pages 85–89, 1999. *Cited on page* 192

[134] D. L. Neuhoff. On the asymptotic distribution of the errors in vector quantization. *IEEE Transactions on Information Theory*, 42(2):461–468, March 1996. *Cited on page* 69

[135] N. C. Oguz and E. Ayanoglu. Performance analysis of two-level forward error correction for lost cell recovery in ATM networks. In *INFOCOM*, pages 728–737, April 1995. *Cited on page* 22

[136] Y. Oohama. The rate-distortion function for the quadratic gaussian ceo problem. *IEEE Transactions on Information Theory*, 44(3):1057–1070, May 1998. *Cited on page* 50

[137] J. M. Ooi and G. W. Wornell. Fast iterative coding techniques for feedback channels. *IEEE Transactions on Information Theory*, 44(7):2960–2976, November 1998. *Cited on page* 85, 89, 92

[138] A. Oppenheim, J. Lim, G. Kopec, and S. Pohlig. Phase in speech and pictures. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, pages 632–637, 1979. *Cited on page* 37

[139] P. Oswald and A. Shokrollahi. Capacity-achieving sequences for the erasure channel. *IEEE Transactions on Information Theory*, 48(12):3017–3028, December 2002. *Cited on page* 80

[140] L. Ozarow. On a Source Coding Problem with Two Channels and Three Receivers. *Bell Syst. Tech. J.*, 59(10):1909–1921, December 1980. *Cited on page* 161

[141] L. H. Ozarow, S. Shamai, and A. D. Wyner. Information theoretic considerations for cellular mobile radio. *IEEE Transactions On Information Theory*, 43(2):359–378, May 1994. *Cited on page* 22, 96, 97, 102, 105, 151

[142] L. H. Ozarow, S. Shamai (Shitz), and A. D. Wyner. Information Theoretic Considerations for Cellular Mobile Radio. *IEEE Trans. Veh. Technol.*, 43(5):359–378, May 1994. *Cited on page* 164

[143] V. Paxson. End-to-end internet packet dynamics. *IEEE Transactions On Networking*, 7(3):277–292, June 1999. *Cited on page* 102

[144] S. S. Pradhan. Source coding with feedforward: Gaussian sources. In *Proc. International Symposium on Information Theory*, Chicago, IL, 2004. *Cited on page* 21, 84, 85, 86

[145] S. S. Pradhan, J. Chou, and K. Ramchandran. Duality between source coding and channel coding and its extension to the side information case. *IEEE Transactions on Information Theory*, 49(5):1181–1203, May 2003. *Cited on page* 84

[146] S. S. Pradhan, J. Kusuma, and K. Ramchandran. Distributed compression in a dense microsensor network. *Signal Processing Magazine*, 19(2):51–60, March 2002. *Cited on page* 50

[147] S. S. Pradhan, R. Puri, and K. Ramchandran. n-Channel Symmetric Multiple Descriptions, Part I: (n,k) Source-Channel Erasure Codes. *IEEE Trans. Inform. Theory*, June 2001. Submitted for publication. Available online at: `http://www.eecs.umich.edu/~pradhanv/paper/ittrans03_4.ps`. *Cited on page* 158, 161, 192

[148] S. S. Pradhan, R. Puri, and K. Ramchandran. n-Channel Symmetric Multiple Descriptions, Part II: An Achievable Rate-Distortion Region. *IEEE Trans. Inform. Theory*, March 2003. Submitted for publication. *Cited on page* 158, 161, 192

[149] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (discus): design and construction. *IEEE Transactions on Information Theory*, 49(3):626–643, March 2003. *Cited on page* 50

[150] V. V. Prelov and E. C. van der Meulen. An asymptotic expression for the information and capacity of a multidimensional channel with weak input signals. *IEEE Transactions on Information Theory*, 39(5):1728–1735, September 1993. *Cited on page* 206

[151] J. G. Proakis and M. Salehi. *Communication Systems Engineering*. Prentice-Hall, Englewood Cliffs, New Jersey, 1994. *Cited on page* 98

[152] R. Puri, S. Pradhan, and K. Ramchandran. n-channel multiple descriptions: theory and constructions. In *Data Compression Conference*, pages 262–271, 2002. *Cited on page* 192

[153] S. Ragot, M. Xie, and R. Lefebvre. Near-ellipsoidal Voronoi coding. *IEEE Transactions on Information Theory*, 49(7):1815–1820, July 2003. *Cited on page* 55

[154] A. R. Reibman, H. Jafarkhani, Y. Wang, M. T. Orchard, and R. Puri. Multiple-description video coding using motion-compensated temporal prediction. *IEEE Transactions On Circuits and Systems for Video Technology*, 12(3):193–204, March 2002. *Cited on page* 158, 161

[155] Z. Reznic, R. Zamir, and M. Feder. Joint source-channel coding of a gaussian mixture source over the gaussian broadcast channel. *IEEE Transactions on Information Theory*, 48(3):776–781, Mar 2002. *Cited on page* 158

[156] B. Rimoldi. Successive Refinement of Information: Characterization of the Achievable Rates. *IEEE Trans. Inform. Theory*, 40(1):253–259, January 1994. *Cited on page* 161

[157] A. Sahai. *Any-time Information Theory.* PhD thesis, Massachusetts Institute of Technology, September 2000. *Cited on page* 84, 114, 190

[158] A. Sahai. Evaluating channels for control: capacity reconsidered. In *American Control Conference*, volume 4, pages 2358–2362, 2000. *Cited on page* 140

[159] D. Sakrison. Worst sources and robust codes for difference distortion measures. *IEEE Trans. Inform. Theory*, 21(3):301–309, May 1975. *Cited on page* 168

[160] J. P. M. Schalkwijk. A coding scheme for additive noise channels with feedback–II: Band-limited signals. *IEEE Transactions on Information Theory*, 12(2):183–189, April 1966. *Cited on page* 85

[161] S. Shamai (Shitz). A Broadcast Strategy for the Gaussian Slowly Fading Channel. In *Proc. International Symposium on Information Theory*, page 150, Ulm, Germany, June 29 – July 4 1997. *Cited on page* 158

[162] C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, pages 623–656, October 1948. *Cited on page* 96

[163] N. Shulman and M. Feder. Static broadcasting. In *Proc. International Symposium on Information Theory*, page 23, 2000. *Cited on page* 25, 98, 138

[164] N. Shulman and M. Feder. Static broadcasting. In *Proc. International Symposium on Information Theory*, page 23, June 2000. *Cited on page* 25, 138, 140

[165] N. J. A. Sloane. Tables of sphere packings and spherical codes. *IEEE Transactions on Information Theory*, 27(3):327–338, May 1981. *Cited on page* 55

[166] R. Smarandache, H. Luerssen, and J. Rosenthal. Constructions of mds-convolutional codes. *IEEE Transactions on Information Theory*, 47(5):2045–2049, July 2001. *Cited on page* 146

[167] X. Tang and A. Zakhor. Matching pursuits multiple description coding for wireless video. *IEEE Transactions On Circuits and Systems for Video Technology*, 12(6):566–575, June 2002. *Cited on page* 158, 161

[168] V. Tarokh and A. Vardy. Upper bounds on trellis complexity of lattices. *IEEE Transactions on Information Theory*, 43(4):1294–1300, July 1997. *Cited on page* 55

[169] S. Tatikonda. *Control Under Communication Constraints*. PhD thesis, Massachusetts Institute of Technology, September 2000. *Cited on page* 84

[170] S. Tatikonda, A. Sahai, and S. K. Mitter. Control of LQG systems under communication constraints. In *IEEE Conference on Decision and Control*, volume 1, pages 1165–1170, December 1998. *Cited on page* 190

[171] R. Vafin and W. Kleijn. Entropy-constrained polar quantization: theory and an application to audio coding. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, pages 1837–1840, 2002. *Cited on page* 37

[172] V. A. Vaishampayan. Design of Multiple Description Scalar Quantizers. *IEEE Trans. Inform. Theory*, 39(3):821–834, May 1993. *Cited on page* 158, 161

[173] R. Venkataramani, G. Kramer, and V. Goyal. Multiple Description Coding with Many Channels. *IEEE Trans. Inform. Theory*, 49(9), September 2003. Accepted for publication. Available online at: `http://cm.bell-labs.com/cm/ms/who/gkr/Papers/mdIT03.pdf`. *Cited on page* 158, 161

[174] R. Venkataramani, G. Kramer, and V. K. Goyal. Bounds on the Achiveable Rate Region for Certain Multiple Description Coding Problems. In *Proc. International Symposium on Information Theory*, page 148, Washington, DC, June 24 - July 29 2001. *Cited on page* 158, 161

[175] H. Viswanathan and T. Berger. The quadratic gaussian ceo problem. *IEEE Transactions on Information Theory*, 43(5):1549–1559, September 1997. *Cited on page* 50

[176] H. Viswanathan and R. Zamir. On the whiteness of high-resolution quantization errors. *IEEE Transactions on Information Theory*, 47(5):2029–2038, July 2001. *Cited on page* 69

[177] E. Viterbo and E. Biglieri. Computing the Voronoi cell of a lattice: the diamond-cutting algorithm. *IEEE Transactions on Information Theory*, 42(1):161–171, January 1996. *Cited on page* 55

[178] E. Viterbo and J. Bouros. A universal lattice code decoder for fading channels. *IEEE Transactions on Information Theory*, 45(5):1639–1642, July 1999. *Cited on page* 66

[179] Y. Wang and S. Lin. Error-resilient video coding using multiple description motion compensation. *IEEE Transactions On Circuits and Systems for Video Technology*, 12(6):438–452, June 2002. *Cited on page* 158, 161

[180] Y. Wang, M. T. Orchard, V. Vaishampayan, and A. R. Reibman. Multiple description coding using pairwise correlating transforms. *IEEE Transactions on Image Processing*, 10(3):351–366, March 2001. *Cited on page* 158, 161

[181] Z. Wang and G. B. Giannakis. What Determines Average and Outage Performance in Fading Channels? In *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, Taipei, Taiwan, November 2002. *Cited on page* 176, 180

[182] P. Whiting and E. Yeh. Optimal encoding over uncertain channels with decoding delay constraints. In *Proceedings of the International Symposium On Information Theory*, page 430, 2000. *Cited on page* 22

[183] H. Witsenhausen. On Source Networks with Minimal Breakdown Degradation. *Bell Syst. Tech. J.*, 59(6):1083–1087, July-August 1980. *Cited on page* 161

[184] H. Witsenhausen and A. D. Wyner. Source Coding for Multiple Descriptions II: A Binary Source. Bell Labs Tech. Rept. TM-80-1217, December 1980. *Cited on page* 158, 161

[185] J. Wolf, A. Wyner, and J. Ziv. Source Coding for Multiple Descriptions. *Bell Syst. Tech. J.*, 59(8):1417–1426, October 1980. *Cited on page* 161

[186] J. K. Wolf and J. Ziv. Transmission of noisy information to a noisy receiver with minimum distortion. *IEEE Transactions on Information Theory*, 16:406–411, July 1970. *Cited on page* 38

[187] J. Wolfowitz. *Coding theorems of information theory.* Springer-Verlag, Berlin/Heidelberg, 3 edition, 1979. *Cited on page* 24, 97, 138

[188] W. S. Wong and R. W. Brockett. Systems with finite communication bandwidth constraints. I. state estimation problems. *IEEE Transactions on Automatic Control*, 42(9):1294–1299, September 1997. *Cited on page* 84

[189] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, IT-22(1):1–10, January 1976. *Cited on page* 21, 84, 85

[190] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22:1–10, January 1976. *Cited on page* 30, 31, 33, 34, 35, 198, 200

[191] M. Yajnik, S. Moon, J. Kurose, and D. Towsley. Measurement and modelling of the temporal dependence in packet loss. In *Proceedings of The Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99)*, pages 345–352, 1999. *Cited on page* 102

[192] Y. Yamada, S. Tazaki, and R. Gray. Asymptotic performance of block quantizers with difference distortion measures. *IEEE Transactions on Information Theory*, 26(1):6–14, January 1980. *Cited on page* 69

[193] R. Zamir. The rate loss in the Wyner-Ziv problem. *IEEE Transactions on Information Theory*, 42(6):2073–2084, November 1996. *Cited on page* 42, 46, 47, 202

[194] R. Zamir. Gaussian codes and shannon bounds for multiple descriptions. *IEEE Transactions on Information Theory*, 45(7):2629–2636, November 1999. *Cited on page* 158, 161, 167, 168

[195] R. Zamir. The half a bit loss of robust source/channel codebooks. In *Proc. Information Theory Workshop*, pages 123–126, Bangalore, India, October 2002. *Cited on page* 35

[196] R. Zamir and U. Erez. A gaussian input is not too bad. *IEEE Trans. Inform. Theory*, 50(6):1362–1367, June 2004. *Cited on page* 176

[197] R. Zamir and M. Feder. On universal quantization by randomized uniform/lattice quantizers. *IEEE Transactions on Information Theory*, 38(2):428–436, March 1992. *Cited on page* 55, 68

[198] R. Zamir and M. Feder. Information rates of pre/post-filtered dithered quantizers. *IEEE Transactions on Information Theory*, 42(5):1340–1353, September 1996. *Cited on page* 55

[199] R. Zamir and M. Feder. On lattice quantization noise. *IEEE Transactions on Information Theory*, 42(4):1152–1159, July 1996. *Cited on page* 69

[200] R. Zamir, S. Shamai, and U. Erez. Nested linear/lattice codes for structured multiterminal binning. *IEEE Transactions on Information Theory*, 48(6):1250–1276, June 2002. *Cited on page* 55

[201] Z. Zhang and T. Berger. New Results in Binary Multiple Descriptions. *IEEE Trans. Inform. Theory*, 33(4):502–521, July 1987. *Cited on page* 158, 161

[202] Z. Zhang and T. Berger. Multiple Description Source Coding with No Excess Marginal Rate. *IEEE Trans. Inform. Theory*, 41(2):349–357, March 1995. *Cited on page* 158, 161

[203] Q. Zhao and M. Effros. Lossless and Near-Lossless Source Coding for Multiple Access Networks. *IEEE Trans. Inform. Theory*, 49(1):112–128, January 2003. *Cited on page* 158

[204] L. Zheng and D. N. C. Tse. Diversity and Multiplexing: A Fundamental Tradeoff in Multiple-Antenna Channels. *IEEE Trans. Inform. Theory*, 49(5):1073–1096, May 2003. *Cited on page* 22, 171, 181