UNIVERSITY OF CALIFORNIA SAN DIEGO

Channel Coding Techniques for Communication over Networks and over Channels with Memory

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Communication Theory and Systems)

by

Nadim Ghaddar

Committee in charge:

Professor Young-Han Kim, Chair
Professor Laurence B. Milstein, Co-Chair
Professor Alireza Salehi Golsefidy
Professor Arya Mazumdar
Professor Paul H. Siegel

2022

The dissertation of Nadim Ghaddar is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

To my parents, sister and brother
whose unconditional love and support
made this dissertation possible.

EPIGRAPH

Next to the originator of a good sentence is the first quoter of it.
I hate quotations. Tell me what you know.
—Ralph Waldo Emerson

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Preparing for this dissertation over the past six years has been a humbling experience. I learned a lot, and unlearned a lot. And for every single part in this process, I had to rely on the support from others, and now is the right time to acknowledge that.

Little can language help me in describing how grateful and honored I am to be a student of Prof. Young-Han Kim. Young-Han not only taught me how to approach a research problem, but he also shaped my research interests. I came to UCSD six years ago with a very blurry vision of what research problems I'd like to work on, and how to work on them. Young-Han, in a way, unclouded that vision. In this process, he showed me how to think clearly, how to write precisely, how to prove rigorously, and how to ask the right questions along the way. He challenged me at the times when I was almost giving up, and when I did, he came to the rescue with new ideas and research avenues for me to explore. On a technical level, Young-Han taught me everything I know about information theory, and did it in the most thought-provoking and resourceful way. His commitment to working on research problems that have both strong theoretical and practical foundations has been inspiring to me; I hope I can approach this dichotomy in a similar way during my career. Young-Han, thank you for the mentor and role model that you have been to me (and will continue to be). Your impact goes way far beyond this PhD.

I am also grateful to Prof. Larry Milstein for his continued support throughout my PhD. I collaborated with Prof. Milstein in my first project on joint channel estimation and polar coding. His insights on the communication theory aspect of the problem were crucial for the results to see light. His very detailed comments on my writing and presentation skills were always on point and helped me grow as a technical writer and presenter. At a personal note, I am thankful for his overall positive attitude towards me – and towards life in general – which helped me through the early stages of my PhD. I am also grateful to the committee members Prof. Paul Siegel, Prof. Arya Mazumdar and Prof. Alireza Salehi Golsefidy for graciously accepting to serve on the committee. I want to especially thank Prof. Siegel for being very approachable throughout my PhD, for his support whenever I needed it, and for his meticulously-planned Probabilistic Coding class. I also would like to thank the late Prof. Alexander Vardy for his constructive comments during my qualifying and preliminary examinations, and also for his comprehensive work on polar codes, some of which inspired the results presented in this dissertation.

I am also deeply indebted to Prof. Lele Wang for her generous mentorship throughout my PhD. Without Lele, I would not have made it to this stage. She has been my unofficial co-advisor (she guided me for the most of my PhD), my favorite collaborator (five out of my eight technical papers are based on joint works with her), and, above all, a genuinely kind friend (she's always generous with her advise and empathetic with her words). I am also grateful to Lele's student, Ziao, for an insightful ongoing collaboration.

I am very thankful for my summer internship opportunities at Samsung and Qualcomm.

Specifically, I would like to thank Dr. Hamid Saber and Dr. Jung Hyun Bae from Samsung and Dr. Ari Klein from Qualcomm for being great mentors for me throughout the internship experiences and for allowing me to explore some of my own ideas during the internships. I am thankful for InterDigital and ETRI (Electronics and Telecommunications Research Institute) for their funding which has supported my PhD journey. I would like to especially thank Dr. Liangping Ma from InterDigital for being a friendly collaborator during the early times of my PhD.

San Diego has been home for the last six years, and it is the people I met here that made it a special one. First, thank you to my labmates – Alankrita, Jiun-Ting, Jongha, Pinar and Shouvik – for the friendly work environment throughout most of my PhD. Although COVID came to ruin our daily meetups in the office and the impromptu lunches on campus (some were pretty late:)), I am grateful for their availability in weekly group meetings from which I learned a lot. I would like to especially thank Shouvik for the brotherhood that we shared through most of my PhD, the stimulating collaboration, and the countless food adventures in San Diego. I am also thankful that my time in the group intersected shortly with our senior Fatemeh. Outside of the lab, a special thank you goes to Rohan, Sukanya, Aditya, Anwesan and Shahar for their close friendship throughout my PhD journey. To them, I owe an improved set of skills in Ping-Pong, a lot of fun road trips exploring California and many memorable birthday celebrations. Thank you to Raghu, Yuhan and Muhammad for all the fun dinner nights. Thank you to Ahmed and Mouna for being the *oldest* friends I know (and for always reminding me that I'm a "kiddo" compared to them:)). Thank you to Govind, Sheel and Pranav for the many matches of Ultimate Frisbee (is it really a national sport?). Thank you to Hanwen, Ayush and Vaishakh for friendly encounters on the fourth floor of Atkinson Hall.

The presence of my cousins Manal, Ali and Leila in nearby Anaheim has given a warm home-like feeling to my stay in California. No words can express how grateful I am to have had them close by. Their home has been my go-to place throughout the last six years. A very special thank you goes to Manal for helping me selflessly at every single step of this journey. My experience in the US wouldn't have been nearly as smooth without her support. (As a side note, she is also the best cook that I have ever met:)). Outside of California, I am grateful to the lifelong friendship of Abed, Wajeb, Rami, Mounib, Razan, Kassir, Taha, and Natali, who never failed to keep me going through their friendly banter. A heartfelt thank you goes to Rakshita for being my day-to-day support system for the better part of my PhD.

Last, but definitely not the least, thank you to the people whom I owe everything to. Baba, Mama, Fattouma and Hammoudi, thank you for all your sacrifices that made me who I am. My deepest gratitude is that you are my family. It is to you I dedicate this dissertation.

Chapter 2, in part, is a reprint with permission of the material as it appears in the paper: Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to coding for network communication,” arXiv:2211.07208, November 2022, which is in preparation to be submitted to *IEEE Transactions on Information Theory*. The dissertation author was the primary investigator and author of this paper. This work was supported in part by the Institute for Information & Communication Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-01659, 5G Open Intelligence-Defined RAN (ID-RAN) Technique based on 5G New Radio), in part by the NSERC Discovery Grant No. RGPIN-2019-05448, and in part by the NSERC Collaborative Research and Development Grant CRDPJ 543676-19.

Chapter 3, in part, is a reprint with permission of the material as it appears in the papers: Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to coding for network communication,” arXiv:2211.07208, November 2022, which is in preparation to be submitted to *IEEE Transactions on Information Theory*, and Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to lossy source coding,” in *2022 17th Canadian Workshop on Information Theory (CWIT)*, pp. 45–50, 2022. The dissertation author was the primary investigator and author of these papers. This work was supported in part by the Institute for Information & Communication Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-01659, 5G Open Intelligence-Defined RAN (ID-RAN) Technique based on 5G New Radio), in part by the NSERC Discovery Grant No. RGPIN-2019-05448, and in part by the NSERC Collaborative Research and Development Grant CRDPJ 543676-19.

Chapter 4, in part, is a reprint with permission of the material as it appears in the papers: Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to coding for network communication,” arXiv:2211.07208, November 2022, which is in preparation to be submitted to *IEEE Transactions on Information Theory*, and Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to coding for asymmetric channels and channels with state,” in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 1367–1372, 2021. The dissertation author was the primary investigator and author of these papers. This work was supported in part by the Institute for Information & Communication Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-01659, 5G Open Intelligence-Defined RAN (ID-RAN) Technique based on 5G New Radio), in part by the NSERC Discovery Grant No. RGPIN-2019-05448, and in part by the NSERC Collaborative Research and Development Grant CRDPJ 543676-19.

Chapter 5, in part, is a reprint with permission of the material as it appears in the paper: Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to coding for network communication,” arXiv:2211.07208, November 2022, which is in preparation to be submitted to *IEEE Transactions on Information Theory*. The dissertation author was the primary investigator and author of these papers. This work was supported in part by the Institute for Information & Communication Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-01659, 5G Open Intelligence-Defined RAN (ID-RAN) Technique based on 5G New Radio), in part by the NSERC Discovery Grant No. RGPIN-2019-05448, and in part by the NSERC Collaborative Research and Development Grant CRDPJ 543676-19.

Chapter 6, in part, is a reprint with permission of the material as it appears in the papers: Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, "A Lego-brick approach to coding for network communication," arXiv:2211.07208, November 2022, which is in preparation to be submitted to *IEEE Transactions on Information Theory*, Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, "A Lego-brick approach to lossy source coding," in *2022 17th Canadian Workshop on Information Theory (CWIT)*, pp. 45–50, 2022, and Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, "A Lego-brick approach to coding for asymmetric channels and channels with state," in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 1367–1372, 2021. The dissertation author was the primary investigator and author of these papers. This work was supported in part by the Institute for Information & Communication Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-01659, 5G Open Intelligence-Defined RAN (ID-RAN) Technique based on 5G New Radio), in part by the NSERC Discovery Grant No. RGPIN-2019-05448, and in part by the NSERC Collaborative Research and Development Grant CRDPJ 543676-19.

Chapter 7, in part, is a reprint with permission of the material as it appears in the papers: Nadim Ghaddar, Young-Han Kim, Laurence B. Milstein, Liangping Ma, and Byung K. Yi, "Joint channel estimation and coding over channels with memory using polar codes," in *IEEE Transactions on Communications*, vol. 69, no. 10, pp. 6575-6589, Oct. 2021, and Nadim Ghaddar, Young-Han Kim, Laurence B. Milstein, Liangping Ma, and Byung K. Yi, "Joint channel estimation and error correction for finite-state markov channels using polar codes," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2018. The dissertation author was the primary investigator and author of these papers.

Chapter 8, in part, is a reprint with permission of the material as it appears in the paper: Alankrita Bhatt, Nadim Ghaddar, and Lele Wang, "Polar coding for multiple descriptions using monotone chain rules," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 565-571, Oct 2017. The dissertation author was the primary investigator and author of this paper.

VITA

2014	B. E. in Computer and Communications Engineering *cum laude*, American University of Beirut, Lebanon
2016	M. Sc. in Communication Systems, École Polytechnique Fédérale de Lausanne, Switzerland
2022	Ph. D. in Electrical Engineering (Communication Theory and Systems), University of California San Diego, United States of America

PUBLICATIONS

Alankrita Bhatt, Nadim Ghaddar, and Lele Wang, "Polar coding for multiple descriptions using monotone chain rules," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 565-571, Oct 2017.

Nadim Ghaddar, Young-Han Kim, Laurence B. Milstein, Liangping Ma, and Byung K. Yi, "Joint channel estimation and error correction for finite-state Markov channels using polar codes," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2018.

Nadim Ghaddar, Hamid Saber, Hsien-Ping Lin, Jung Hyun Bae, and Jungwon Lee, "Simplified decoding of polar codes by identifying Reed-Muller constituent codes," in *2020 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2020.

Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, "A Lego-brick approach to coding for asymmetric channels and channels with state," in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 1367–1372, 2021.

Nadim Ghaddar, Young-Han Kim, Laurence B. Milstein, Liangping Ma, and Byung K. Yi, "Joint channel estimation and coding over channels with memory using polar codes," in *IEEE Transactions on Communications*, vol. 69, no. 10, pp. 6575-6589, Oct. 2021.

Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, "A Lego-brick approach to lossy source coding," in *2022 17th Canadian Workshop on Information Theory (CWIT)*, pp. 45–50, 2022.

Ziao Wang, Nadim Ghaddar, and Lele Wang, "Noisy sorting capacity," in *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 2541–2546, 2022.

Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, "A Lego-brick approach to coding for network communication," arXiv:2211.07208, November 2022, in preparation to be submitted to *IEEE Transactions on Information Theory*

ABSTRACT OF THE DISSERTATION

Channel Coding Techniques for Communication over Networks and over Channels with Memory

by

Nadim Ghaddar

Doctor of Philosophy in Electrical Engineering
(Communication Theory and Systems)

University of California San Diego, 2022

Professor Young-Han Kim, Chair
Professor Laurence B. Milstein, Co-Chair

Next-generation wireless communication systems will have to deal with an unprecedented number of communicating users and devices while enabling orders-of-magnitude of performance improvement in speed and connectivity. With the increasingly complex network structure and the high spectral efficiency requirements, it becomes extremely inefficient to rely on traditional channel coding paradigms that do not take into account the structure of the network and its inherent properties. Unlike conventional channel coding schemes that are designed under the assumption of a single sender and a single receiver communicating over a memoryless channel, this dissertation investigates low-complexity channel coding techniques that take advantage of the number of communicating devices in a network and the inherent memory in the channel. In communication over networks, low-complexity channel coding schemes that achieve the best known information theoretic performance are constructed starting from simple coding blocks. In communication over channels with memory, practical channel coding techniques that exploit the memory in the channel are developed. In both cases, the proposed coding techniques have the potential of addressing the increasing-spectral-efficiency requirement in next-generation wireless communication systems.

Chapter 1

Introduction

In his plenary talk in the 2001 IEEE International Symposium on Information Theory, Robert J. McEliece asked: “Are turbo-like codes effective on nonstandard channels?” [1]. At the time, it was well-understood that such codes perform pretty well on binary-input memoryless symmetric channels. Hence, it was natural to investigate how such codes perform in more general settings, e.g., over channels that are non-binary, non-memoryless, non-symmetric, and mutliuser. The basic conclusion that McEliece alluded to is that binary turbo codes [2] and their relatives (e.g., low density parity check codes [3, 4, 5]) have the potential to be used effectively over non-standard channels using low-complexity graph-based iterative decoding. Fast-forward to around a decade later, coding theory saw a couple of extraordinary breakthroughs that are manifested by the invention of polar codes [6] and spatially coupled codes [7]. Not only that these two families of codes achieve the capacity of binary-input memoryless symmetric channels, but they also do so with efficient encoding and decoding algorithms. Therefore, today more than ever, there is a need to investigate the degree to which the results on standard channels can be extended to more general communication scenarios.

Concurrent with the advances in coding theory, the mobile industry is growing at an unprecedented rate. According to Cooper’s law [8], which tracks the performance of wireless communication systems through time, the number of mobile connections per unit area has increased by a factor of 10^6 over the past fifty years. Cooper’s study divides this factor of one million into three parts: a factor of 25 is due to allocation of more spectrum, a factor of 25 is due to improved spectral efficiency of physical layer techniques, and a factor of 1600 is attributed to the use of more base stations in significantly denser spatial configurations. Indeed, the trend of increased deployment of ultra-dense base stations is especially evident in 5G systems, with base stations (cells) being deployed every 100 meters in certain urban areas. There are two major implications of such closely packed base stations. First, the interference between different users and base stations becomes much stronger. This is exacerbated by the expected explosion of the number of communicating users (and devices) due to emerging applications in transportation systems, smart cities, and cloud computing. Second, the channels between the base stations and the users become much more complex and interdependent. Hence, traditional assumptions on underlying channel models such as memorylessness and symmetricity become of little practical value. With such complex network structure, many users and interfering radio, it becomes extremely inefficient to rely on standard channel coding techniques that do not take into account the structure of the networks and their inherent properties.

Inspired by the theoretical advances and the pressing need in practice, this dissertation is an attempt to devise low-complexity channel coding techniques for nonstandard channel models. In particular, we consider channels that involve multiple senders and receivers (i.e., networks), and channels that have memory. Before digging into the details, a brief description of the contributions of this dissertation is given below (please see Section 1.4 for a more elaborate discussion of the contributions).

Coding for Network Communication. Most existing channel coding techniques developed for network communication are either based on dividing the network into separate point-to-point links (i.e., links that involve a single sender and a single receiver) – and hence, are far from being rate-optimal for network communication – or are based on the use of high-complexity encoding and decoding methods such as joint typicality encoding and maximum likelihood decoding – and hence, are not practical. Towards the ambitious goal of closing this gap between what is theoretically known to be optimal for communication over networks and what is practically feasible, we take a *Lego-brick approach* to transform conceptual coding schemes developed in network information theory to practical implementations. In this approach, we identify basic coding blocks for one (or more) communication setting and combine them together to build another coding block for a more complex communication setting[1]. The result of this research thread is a collection of coding schemes for several network scenarios that are both rate-optimal and practical, and that can be implemented starting from existing coding blocks designed for binary-input memoryless symmetric point-to-point channels.

Coding over Channels with Memory. In many practical scenarios, the memory in a channel asserts itself through time-varying channel state parameters that are unknown to both the sender and the receiver. The common paradigm of designing coding schemes for such channels is to assume that the receiver is able to compute accurate estimates of these parameters prior to decoding (e.g., using a known pilot sequence). This separation of channel estimation and decoding is known to be sub-optimal from an information theoretic perspective, since the distribution of the channel state process is ignored in the decoding procedure. Alternatively, instead of estimating the channel states and then performing channel decoding separately, we devise a joint channel estimation and coding scheme that incorporates the tasks of pilot symbol insertion and channel estimation into the encoding and decoding procedures of polar codes over several channel models with memory. The result of this

[1] Can you see the analogy to building Lego's?

work is a polar-coding-based technique to code over channels with memory which requires less pilot overhead and seems particularly suited to be used over fast-varying channels.

To establish the context of these contributions, we first go through a brief history of coding theory in Section 1.1, along with the most important code constructions that are capacity-achieving over point-to-point channels. In Sections 1.2 and 1.3, we briefly motivate the main two themes of this dissertation, namely, coding over networks and coding over channels with memory. Finally, in Section 1.4, we elaborate on the main contributions of this dissertation and its overall outline.

1.1 Road to Channel Capacity

The technology of communication and computing advanced at a breathtaking pace in the twentieth century, particularly after Claude Shannon's landmark paper "A Mathematical Theory of Communication" in 1948 [9], which constituted the birth of the field of *information theory.* In his paper, Shannon posed the fundamental question: what is the maximum rate at which reliable communication is possible, and how can we efficiently communicate close to this rate? He considered the architecture depicted in Figure 1.1, where a sender wishes to communicate a k-bit message M to a receiver over a noisy channel. The message is mapped by an encoder to an n-symbol channel input sequence $X^n(M)$, and the received channel output sequence Y^n is mapped by a decoder to a message estimate $\hat{M}(Y^n)$.

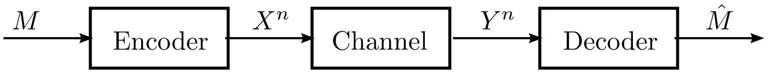


Figure 1.1: Shannon's model of a point-to-point communication system.

Shannon took an asymptotic approach to characterize the necessary and sufficient condition for reliable communication. His ingenious formulation of the point-to-point communication problem led to the following fundamental theorem.

Point-to-pint channel coding theorem. Suppose that the channel is discrete and memoryless with input X, output Y and conditional probability $p(y\,|\,x)$ that specifies the probability of receiving the symbol y when x is transmitted. The decoder wishes to find an estimate $\hat{M}$ of the message with a small probability of decoding error $\mathsf{P}\{\hat{M} \neq M\}$. Shannon formulated the problem as one of finding the *channel capacity* C, which is the maximum rate $R = k/n$ in bits per channel transmission at which the probability of error can be made arbitrarily small when n is sufficiently large. He elegantly characterized the channel capacity as

$$C = \max_{p(x)} I(X;Y) \qquad \text{bits/transmission,} \tag{1.1}$$

where $I(X;Y)$ denotes the mutual information between the channel input X and the channel output Y.

Shannon's construction of an encoder-decoder pair that achieves the capacity of a discrete memoryless channel was based on *random coding.* By looking at an ensemble of codes generated according to the capacity-achieving input distribution (i.e., the one that maximizes (1.1)) and analyzing the error probability of a typicality decoder, Shannon showed that there should exist at least one code in the ensemble that achieves the capacity of the channel. Other classic proofs of Shannon's achievability theorem based on random coding are found in [10, 11, 12]. For a detailed overview of related results in information theory, the interested reader is referred to the standard textbooks in [13] and [14].

Nonetheless, due to their large decoding complexity, random codes are not amenable to practical applications. Therefore, ever since Shannon's 1948 paper, the quest for finding low-complexity codes that approach capacity has been the central objective of *coding theory.* From algebraic constructions (such as Hamming [15], Golay [16], Reed–Muller [17, 18], Bose–Chaudhuri–Hocquenghem [19, 20] and Reed–Solomon codes [21]) to probabilistic constructions (such as turbo [2], low-density parity-check [3, 4, 5], and expander codes [22]), to the polar [6] and spatially coupled codes [7] mentioned earlier, coding theory has made significant strides to approach and later achieve Shannon's fundamental limit of reliable communication over discrete memoryless point-to-point channels. The main code constructions known to achieve Shannon's capacity for point-to-point channels are the following:

Random-coding-based constructions. A random code whose codewords are generated (pairwise) independently according to the capacity-achieving input distribution achieves the capacity of all discrete memoryless channels[2]. Moreover, a random *linear* code achieves the capacity of discrete memoryless *symmetric* channels. However, in both cases, no low-complexity decoding method is known, in general, for these codes.

Spatially coupled codes. Spatially coupled low-density parity-check codes achieve the capacity of all binary-input memoryless symmetric channels. These codes can be decoded efficiently using the iterative message-passing algorithm.

Polar codes. Polar codes achieve the capacity of all binary-input memoryless symmetric channels using the low-complexity successive cancellation decoding algorithm. The achievability proof is based on the intriguing phenomenon of *channel polarization.*

Reed–Muller codes. Reed–Muller codes achieve the capacity of binary erasure channels using maximum-a-posteriori (MAP) decoding [25]. Moreover, Reed–Muller codes have a vanishing *bit-error-rate* over all binary-input memoryless symmetric channels using bitwise maximum-a-posteriori

[2] In fact, it is sufficient that the *weight distribution* of the code (i.e., the distribution of the number of 1's in the codewords) is close to that of the random ensemble to establish capacity achievability [23, 24].

(bit-MAP) decoding [26]. However, in both cases, the decoder has a complexity that is exponential in the block length, and, thus, is not friendly to practical implementations.

1.2 Beyond P2P Channels: Coding over Networks

The simplistic model of a communication system as a single source-destination pair communicating over a noisy channel does not capture many important aspects of real-world networks. Wireless communication systems, for example, use a shared broadcast medium between potentially multiple senders and multiple receivers. Inspired by Shannon's formulation of the point-to-point communication problem, network information theory aims to study the fundamental limits of reliable communication over networks and the optimal coding schemes that can achieve those limits. The codes developed in network information theory require techniques beyond point-to-point channel coding, and can achieve, in general, strictly larger rates compared to the basic approach of coding over the point-to-point links of a network.

Let's consider an example. A two-user Gaussian broadcast channel, which is used to model the downlink of a cellular system, is depicted in Figure 1.2a. The sender wishes to communicate a message M_j to each user j at a rate R_j, for $j = 1, 2$. The channel output at the each user can be expressed as

$$\begin{aligned} Y_1 &= g_1 X + Z_1, \\ Y_2 &= g_2 X + Z_2, \end{aligned}$$

where $Z_1 \sim \mathcal{N}(0, 1)$ and $Z_2 \sim \mathcal{N}(0, 1)$ are independent noise components at the receivers, and $g_1^2 > g_2^2$, that is, the channel seen by receiver 1 is stronger than that seen by receiver 2. Assume that the transmitter is subject to an average transmission power constraint P. One approach to communicate over this channel is to send the messages separately to each user (e.g., in different time intervals or frequency bands) using the point-to-point channel codes discussed previously. In this case, we can reliably communicate at rate pairs (R_1, R_2) in the "time-division region" $\mathcal{R}$ shown in Figure 1.2b, where $C_j = \frac{1}{2}\log(1 + g_j^2 P)$ is the capacity of the point-to-point Gaussian channel $X \to Y_j$ with signal-to-noise ratio $g_j^2 P$, for $j = 1, 2$. Cover [27] showed that a strictly larger rate region can be achieved by adding the codewords for the two messages and sending the sum over the channel. The stronger receiver 1 decodes both codewords, while the weaker receiver 2 treats the other codeword as noise and decodes only its own codeword. This *superposition coding* scheme allows to communicate at any rate pair in the capacity region $\mathcal{C}$ shown in Figure 1.2b, a strictly larger region compared to the time-division region $\mathcal{R}$. Similar improvements in communication rates can be achieved for the uplink (multiple access channel), intercell interference (interference channel) and many other practically-motivated network models. These intriguing results motivate the quest to find practical low-complexity coding schemes for communication over networks. For a detailed overview of results in network information theory, we refer to the textbook [28] which contains an

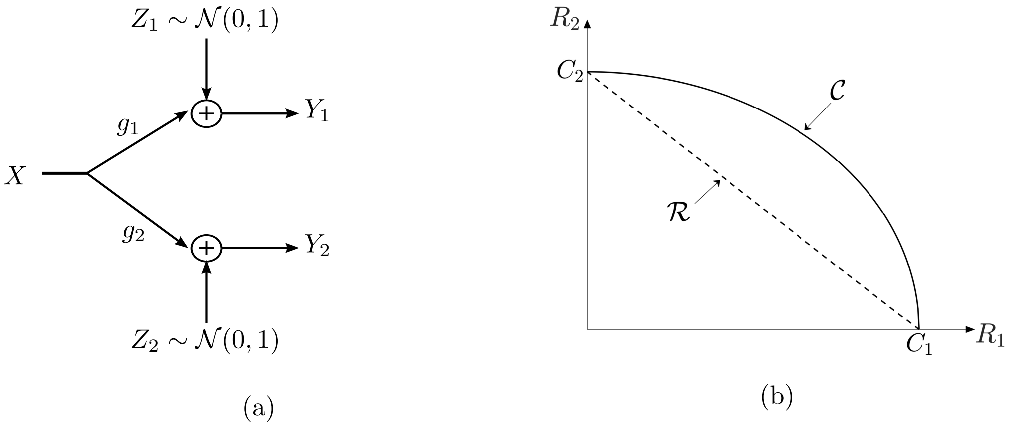


Figure 1.2: (a) Two-user Gaussian broadcast channel. (b) The time-division rate region $\mathcal{R}$ and the capacity region $\mathcal{C}$.

encyclopedic coverage of this area.

1.3 Beyond Memoryless Channels: Coding over Channels with Memory

Most real-world channels have memory. Most prominently, fading channels in wireless communication are time-varying due to multiple signal paths and user mobility, and introduce channel gains that randomly change over time. The traditional solution to deal with channel memory is to interleave the encoded sequence of symbols prior to transmission and to deinterleave the received output sequence prior to decoding[3]. If the interleaving span is long enough, the interleaved channel may be considered memoryless, in which case the point-to-point channel codes discussed earlier can be used. Nonetheless, such an approach is known to be sub-optimal from an information theoretic perspective as the rates that can be achieved over the interleaved channel – under the assumption of no memory – are strictly smaller than the capacity of the original channel [30]. Note that the interleaving process itself does not reduce the capacity of the channel; it is the code which ignores the inherent channel memory that is insufficient to achieve it.

Moreover, when the memory in the channel is manifested through unknown state parameters that change randomly over time (e.g., fading channels), the underlying assumption that governs the design of most existing channel coding techniques is that the receiver is able to estimate the state parameters accurately enough prior to decoding. In most practical systems, a predetermined training sequence (also known as a pilot sequence), containing no actual information, is transmitted to help the receiver in this task. Once estimates of the channel states at the pilot symbol positions

[3] For example, such an interleaving process is used in 5G systems in both the control and data channels [29].

are computed, the receiver interpolates (i.e., "tracks" the channel) to get estimates of the channel states at the data symbol positions, and only then proceeds to perform channel decoding. As mentioned earlier, this separation of estimation and decoding is sub-optimal in general, particularly when the channel states are changing at a fast rate (e.g., in the case of high mobility) [31, 32]. This motivates the quest for finding low-complexity encoders and decoders that take into account the inherent memory in the channel.

1.4 Outline and Contributions of the Dissertaion

This dissertation tells a story about coding over nonstandard channels, in particular, channels with multiple senders and receivers (i.e., networks) and channels with memory. Let us first give a bird's eye view on the structure of this thesis. The topic of Chapters 2 to 6 is the Lego-brick approach to coding over networks: in Chapter 2, we motivate this approach and describe necessary tools that will be used in subsequent chapters; in Chapter 3, we talk about source coding problems with multiple terminals; in Chapter 4, we discuss channel coding problems over networks; Chapter 5 talks about coding over cloud radio access networks, and Chapter 6 describes modified coding schemes based on block-Markov coding. Since the coding schemes presented in these chapters become more involved from one chapter to the other, it is recommended that Chapters 2 to 6 are read in the order that they are presented. In Chapter 7, we talk about channels with memory and describe a joint channel estimation and polar coding scheme over these channels. In Chapter 8, we describe a polar coding scheme for the multiple description coding problem. Each of Chapters 7 and 8 is self-contained, and can be read independently from other chapters.

1.4.1 Coding Over Networks: A Lego-Brick Approach

Our first contribution is a systematic framework for designing coding schemes for network communication [33]. Starting from point-to-point channel codes that are designed for symmetric channels (such as the ones mentioned in Section 1.1), we identify basic properties that these codes should satisfy so that they can be used in the construction of a coding scheme for a given network information theory problem. Viewing the channel codes that satisfy those properties as "black boxes", we will provide guidelines onto how to assemble them together in order to construct a practical coding scheme for a given network problem. We will refer to such an approach to coding as a "Lego-brick approach" and to the constituent channel codes as "Lego bricks", taking a literal analogy to building complex Lego objects starting from simple components. In particular, we will construct coding schemes for the problems of lossless source coding [9], Slepian–Wolf coding [34], lossy source coding [35], Wyner–Ziv coding [36], Gelfand–Pinkser coding [37], asymmetric channel coding [9], multiple description coding [38], Berger–Tung coding [39], coding for multiple access channels [40], Marton coding for broadcast channels [41], and coding for cloud radio access networks

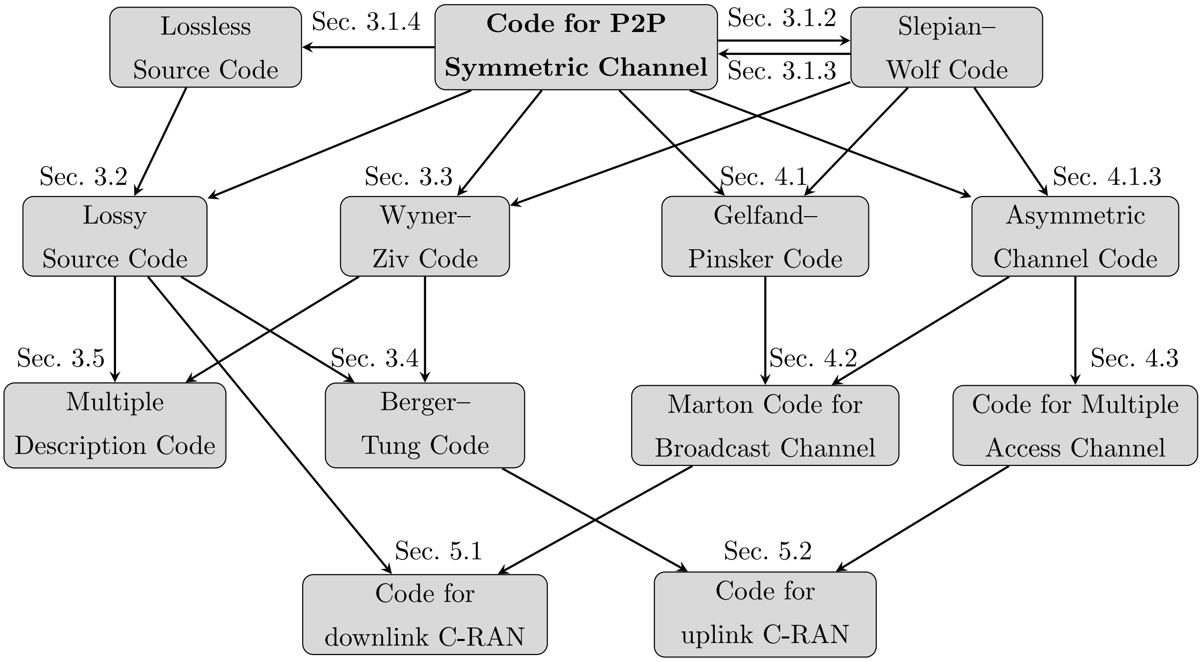


Figure 1.3: Overview of the proposed coding schemes and their constituent building blocks, along with the section number in which each coding scheme is presented.

(C-RAN's) [42]. The code constructions are friendly to practical implementation and can achieve rate regions that are strictly larger compared to the naive approach of coding over the point-to-point links of a network. In fact, for most of the problems that we consider, the achieved rate regions are the best known inner bounds[4], provided that the constituent Lego bricks are rate-optimal.

Figure 1.3 illustrates the different coding problems considered in this dissertation, along with their constituent Lego bricks. Arrows pointing from a set of coding problems towards another coding problem means that codes for these problems can be used as building blocks in the design of a code for the designated problem. More specifically, coding schemes for the following problems will be constructed starting from the following codes.

1. Code for point-to-point symmetric channel → Lossless source code

2. Code for point-to-point symmetric channel → Slepian–Wolf code

3. Slepian–Wolf code → Code for point-to-point symmetric channel

4. Code for point-to-point symmetric channel + Lossless source code → Lossy source code

5. Code for point-to-point symmetric channel + Slepian–Wolf code → Wyner–Ziv code

[4] The exceptions are the downlink (uplink) C-RAN problems, where the rate regions achieved through our constructions can be improved through the high-complexity joint encoding (decoding) of the user messages.

6. Code for point-to-point symmetric channel + Slepian–Wolf code → Gelfand–Pinsker code
7. Code for point-to-point symmetric channel + Slepian–Wolf code → Asymmetric channel code
8. Two lossy source codes + Wyner–Ziv code → Multiple description code
9. Lossy source code + Wyner–Ziv code → Berger–Tung code
10. Asymmetric channel code + Gelfand–Pinsker code → Marton code for broadcast channel
11. Two asymmetric channel codes → Code for the two-user multiple access channel
12. Two lossy source codes + Marton code for broadcast channel → Code for the downlink C-RAN problem
13. Multiple access channel code + Berger–Tung code → Code for the uplink C-RAN problem

Therefore, all the coding schemes can be constructed starting from point-to-point channel codes designed for symmetric channels.

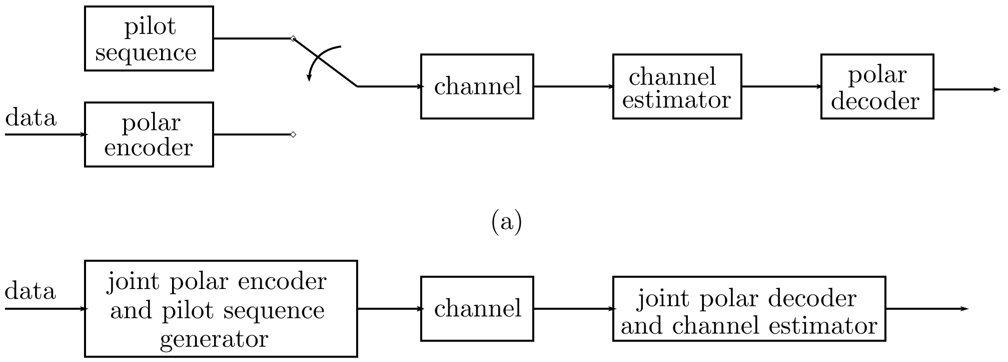
In **Chapter 2**, we formally define the point-to-point channel coding problem and describe the primitive properties that point-to-point channel codes should satisfy so that they can be used as building blocks in a coding scheme for network communication. We will also describe in detail the motivations behind the Lego-brick approach and the related work in the literature.

In **Chapter 3**, we discuss the source coding problems, namely, the problems of lossless source coding, Slepian–Wolf coding, lossy source coding, Wyner–Ziv coding, Berger–Tung coding and multiple description coding. For each coding problem, we give explicit constructions of coding schemes starting from basic building blocks and analyze the performance. We also provide simulation results for the lossy source coding problem.

In **Chapter 4**, we proceed to the channel coding problems, namely, the problems of Gelfand–Pinsker coding, asymmetric channel coding, Marton coding over broadcast channels and coding over multiple access channels. For each coding problem, explicit constructions are given and the performance of the coding schemes is analyzed. Simulation results are provided for Gelfand–Pinsker coding and Marton coding.

In **Chapter 5**, we describe coding schemes for cloud radio access networks (uplink and downlink). These networks, by nature, involve both source and channel coding counterparts, and, hence, are treated in a separate chapter. Indeed, the constructions for these networks will use tools from the previous two chapters. Simulation results are provided for the both the uplink and downlink scenarios.

In **Chapter 6**, we describe modified constructions for all the previous coding schemes based on block-Markov coding. Unlike the previous constructions, the coding schemes described in this chapter involve properties of the Lego bricks that can be easily verified in practice for any off-the-shelf code. Unfortunately, this comes at the cost of a larger implementation complexity, a


pilot sequence
data
polar encoder
channel
channel estimator
polar decoder
data
joint polar encoder and pilot sequence generator
channel
joint polar decoder and channel estimator

(a)

(b)

Figure 1.4: (a) Separate channel estimation and coding. The transmitter alternates between sending pilot sequences and coded data, and the receiver performs estimation and decoding separately. (b) Joint channel estimation and coding.

small penalty incurred in the achievable rates, and a worse performance guarantee. The coding schemes have a block-Markov structure, i.e., the input to one coding block might depend on the inputs/outputs of previous coding blocks.

1.4.2 Joint Estimation and Coding over Channels with Memory

Our second main contribution is a joint channel estimation and polar coding scheme for channels with memory [43]. Unlike the conventional approach of first estimating all channel parameters and then performing channel decoding separately (Figure 1.4a), the proposed scheme incorporates a subset of reliable estimates of channel parameters into the decoding procedure and computes decoding metrics averaged over the statistical behavior of the channel (Figure 1.4b). Moreover, the structure of polar codes is exploited to develop a pilot arrangement scheme that embeds pilot symbols within the polar codewords. The contributions of this part can be summarized by the following.

1. A polar decoding algorithm for finite-state Markov channels of any order. An "estimation-aware" variant of the algorithm that computes decoding metrics conditioned on channel estimates is also presented.

2. A successive cancellation decoder for the Gauss-Markov channel. As the channel state alphabet is not finite in this case, the decoder is not a straightforward extension from that of finite-state Markov channels.

3. An iterative estimation-decoding scheme that incorporates reliably-decoded bits into subsequent iterations of channel estimation. The scheme uses list decoding of polar codes to identify

the reliably-decoded bits.

4. A pilot arrangement scheme that uses the special encoding structure of polar codes to align known pilot symbols with code bits. Equivalent to the shortening of a polar code, the scheme improves on existing shortening schemes in the literature by adding flexibility in choosing the positions of the shortened bits.

The simulation results over finite-state Markov channels, Gauss-Markov channels and flat-fading channels demonstrate significant performance gains compared to separate estimation and coding. These contributions will be the subject of **Chapter 7**.

1.4.3 Polar Codes for Multiple Description Coding

Our third main contribution is a polar coding scheme for the multiple description coding problem that can achieve the entire El Gamal–Cover inner bound for this problem [44]. A key ingredient in this result is exploiting an interesting duality between the multiple description coding problem and the multiple access channel coding problem. The coding scheme we develop in this part is inherently different than our previous Lego-brick design (see multiple description coding in Figure 1.3). In particular, the proof technique here is specific to polar codes, and the construction can achieve any point on the dominant face of the El Gamal–Cover rate region (not just a corner point). Due to these differences, this result will be addressed separately and will be the subject of **Chapter 8**.

Chapter 2

Towards a Lego-Brick Approach to Coding

In this chapter, we build towards a unified framework for coding over networks starting from simple coding blocks. We start with defining the point-to-point channel coding problem in Section 2.1. In Section 2.2, we define the notion of a symmetrized channel, which will turn out to be crucial in most of our code constructions over networks. In Section 2.3, we identify two properties of point-to-point channel codes that allow to translate their performance to network settings, namely, the error probability and the decoding distance. Sections 2.4 and 2.5 are devoted to motivate the Lego-brick approach to coding and describe the related previous work, respectively. We start from point-to-point channel coding.

2.1 Point-to-point Channel Codes: A Formal Definition

Consider a binary-input discrete memoryless channel $p(y\,|\,x)$ with an input alphabet $\mathcal{X} = \{0,1\}$, an output alphabet $\mathcal{Y}$ and a collection of conditional probability mass functions (pmf's) $p(y\,|\,x)$ on $\mathcal{Y}$ for each $x \in \mathcal{X}$. A (k,n) point-to-point channel code (f,ϕ) for the channel $p(y\,|\,x)$ consists of

- a codebook $\mathcal{C} \subseteq \{0,1\}^n$ of size $|\mathcal{C}| = 2^k$,
- an encoder $f : [2^k] \to \mathcal{C}$ that maps each message $m \in [2^k]$ to a codeword $x^n = f(m)$,
- a decoder $\phi : \mathcal{Y}^n \to \mathcal{C}$ that assigns a codeword estimate $\hat{x}^n = \phi(y^n)$ to each received sequence y^n.

The rate of the code is $R = k/n$. We say that the channel code is *linear* if for any two codewords $c^n, \tilde{c}^n \in \mathcal{C}$, we have $c^n \oplus \tilde{c}^n \in \mathcal{C}$. A linear code can be alternatively defined by its parity-check matrix

$H_{n-k\times n}$ and its decoding function ϕ. In this case, the codebook can be written as $\mathcal{C} = \{c^n : Hc^n = 0^{n-k}\}$. With a slight abuse of notation, when the code is linear and its parity-check matrix is H, we will refer to it as a (k, n) linear point-to-point channel code (H, ϕ).

Remark 2.1.1. Given any $(n-k) \times n$ parity-check matrix H, we will assume throughout this dissertation that the last $n-k$ columns of H are linearly independent, i.e., $H = \begin{bmatrix} A & B \end{bmatrix}$ for some nonsingular $(n-k) \times (n-k)$ matrix B. We also introduce the following linear transformation of the parity-check matrix,

$$\widetilde{H} = \begin{bmatrix} \mathbf{0} \\ B^{-1}H \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ B^{-1}A & I \end{bmatrix}, \tag{2.1}$$

where $\mathbf{0}$ denotes the all-zero matrix of the appropriate dimension.

Definition 2.1.1 (BMS Channel)**.** We say that a binary-input memoryless channel $p(y\,|\,x)$ is *symmetric* (abbreviated, a BMS channel) if there exists a permutation $\pi : \mathcal{Y} \to \mathcal{Y}$ such that $\pi^{-1} = \pi$ and $p(y\,|\,x) = p(\pi(y)\,|\,x \oplus 1)$ for all $y \in \mathcal{Y}$ and $x \in \{0, 1\}$. The channel $p(y\,|\,x)$ is *asymmetric* if it is not symmetric.

Remark 2.1.2. The capacity-achieving input distribution for a BMS channel is the uniform distribution [45, Theorem 4.5.2]. Note that this is the only input distribution that can be attained using linear code ensembles.

2.2 Symmetrized Channel

Given any binary-input channel $p(y\,|\,x)$ (that is not necessarily symmetric) and any input distribution $p(x)$ (that is not necessarily uniform), a technique that will be crucial for us in our code constructions is the concept of a *symmetrized channel* corresponding to the joint distribution $p(x, y) = p(x)p(y\,|\,x)$, defined as follows [46].

Definition 2.2.1 (Symmetrized Channel)**.** Given a binary-input channel $p(y\,|\,x)$ and an input distribution $p(x)$ (not necessarily uniform), the *symmetrized channel* corresponding to $p(x, y) = p(x)p(y\,|\,x)$ is defined as the channel $\bar{p}$ with input alphabet $\mathcal{X} = \{0, 1\}$, output alphabet $\mathcal{Y} \times \{0, 1\}$ and transition probabilities

$$\bar{p}(y, v\,|\,x) = p_{X,Y}(x \oplus v, y).$$

Remark 2.2.1. The following are immediate consequences of the definition of a symmetrized channel.

(i) The channel $\bar{p}$ is symmetric under permutation $\pi\big((y, v)\big) = (y, v \oplus 1)$. In other words, for any $s \in \{0, 1\}$,

$$\bar{p}(y, v\,|\,x) = \bar{p}(y, v \oplus s\,|\,x \oplus s),$$

which extends naturally when considering length-n sequences.

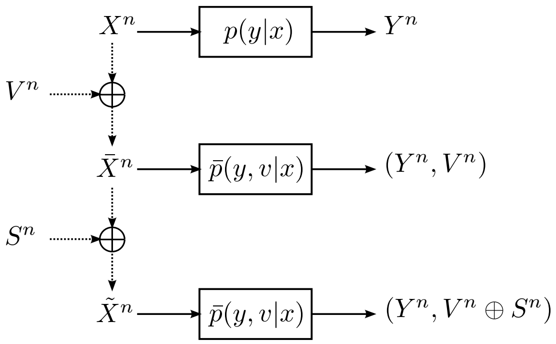


Figure 2.1: The symmetrized channel.

(ii) The channel $\bar{p}$ is, in particular, the conditional distribution $p_{\bar{Y}|\bar{X}}$ when $\bar{X} = X \oplus V$, $\bar{Y} = (Y, V)$, (X, Y) is distributed according to $p(x, y)$, and $V \sim \text{Bern}(1/2)$ is independent of (X, Y).

(iii) Since $\bar{X} \sim \text{Bern}(1/2)$ and the channel $\bar{p}$ is symmetric, it follows that the capacity of the symmetrized channel $\bar{p}$ is

$$I(\bar{X}; \bar{Y}) = H(\bar{X}) - H(\bar{X} \,|\, \bar{Y}) = 1 - H(X \oplus V \,|\, Y, V) = 1 - H(X \,|\, Y).$$

Observations (i) and (ii) made above are illustrated in Figure 2.1. If (X^n, Y^n) are i.i.d. according to $p(x, y)$, then the symmetrized channel $\bar{p}$ is exactly the channel between $\bar{X}^n = X^n \oplus V^n$ and (Y^n, V^n), where V^n is an i.i.d Bern(1/2) sequence that is independent of (X^n, Y^n). Furthermore, the symmetric property of the channel $\bar{p}$ implies that, for any arbitrary sequence S^n, the channel between $\tilde{X}^n = \bar{X}^n \oplus S^n$ and $(Y^n, V^n \oplus S^n)$ is also described by $\bar{p}$. This property will turn out to be useful in several of our constructions in the coming chapters.

2.3 Two Primitive Properties of Point-to-Point Codes

Given a (k, n) linear point-to-point channel code (H, ϕ) designed for a BMS channel $p(y \,|\, x)$, we will focus throughout this dissertation on two properties of the code, namely, the *error probability* and the *decoding distance*, which we define precisely below.

(1) Error probability: Let $X^n \sim \text{Unif}(\mathcal{C})$, where $\mathcal{C}$ is the codebook corresponding to H, and let Y^n be the output of the channel $p(y \,|\, x)$ when the input is X^n. The error probability ϵ of the code (H, ϕ) when used over the channel $p(y \,|\, x)$ is defined as the probability of decoding error, i.e.,

$$\epsilon \triangleq \sum_{x^n \in \mathcal{C}} \Big(2^{-k}\, \mathsf{P}\{\phi(Y^n) \neq X^n \mid X^n = x^n\} \Big).$$

Shannon's point-to-point channel coding theorem [9], along with its achievability proof using linear codes [12], state that a sequence of (nR, n) linear codes having a vanishing error probability over the BMS channel $p(y\,|\,x)$ exists if and only if the rate $R < I\big(\mathrm{Bern}(1/2), p(y\,|\,x)\big)$, where $I\big(\mathrm{Bern}(1/2), p(y\,|\,x)\big)$ means the mutual information $I(X;Y)$ of the channel $p(y|x)$ when the input $X \sim \mathrm{Bern}(1/2)$. Note that since the channel is symmetric, $I\big(\mathrm{Bern}(1/2), p(y\,|\,x)\big)$ is, in fact, the capacity of the channel.

(2) Decoding distance: Let Y^n be an i.i.d. $p(y)$ sequence, where $p(y) \triangleq \sum_x \frac{1}{2} p(y\,|\,x)$ is the marginal output distribution when the input distribution is Bern(1/2). Define $X^n = \phi(Y^n)$. The decoding distance δ of the code (H, ϕ) with respect to the channel $p(y\,|\,x)$ is defined as the total variation distance between the distribution of (X^n, Y^n) (let's denote that by $q(x^n, y^n)$) and the i.i.d. $\frac{1}{2} p(y\,|\,x)$ distribution, i.e.,

$$\delta \triangleq \frac{1}{2} \sum_{x^n, y^n} \left| q(x^n, y^n) - \frac{1}{2^n} \prod_{i=1}^{n} p(y_i\,|\,x_i) \right| .$$

The decoding distance can be understood as a measure of the stochastic behavior of the decoding function ϕ in comparison to the "backward" channel $p(x\,|\,y)$ corresponding to the capacity-achieving input distribution. A sequence of (nR, n) linear codes with a vanishing decoding distance exists if and only if the rate $R > I\big(\mathrm{Bern}(1/2), p(y\,|\,x)\big)$. To see this, one can refer to results on the distributed channel synthesis problem, introduced by Bennett *et al.* in [47] and further characterized by Cuff in [48]. In this problem, an i.i.d. source $\bar{Y}^n$ distributed according to $p(\bar{y})$ is encoded by an index $M \in [2^{nR_{\mathrm{s}}}]$ to a decoder that wishes to produce an output $\bar{X}^n$ such that the joint distribution of $(\bar{X}^n, \bar{Y}^n)$ is indistinguishable (in total variation distance) from a given joint i.i.d distribution $p(\bar{x}, \bar{y})$. This is referred to as "synthesizing" the channel $p(\bar{x}\,|\,\bar{y})$. The main result in the distributed channel synthesis literature [48] is that there exists a construction of an encoder-decoder pair to synthesize $p(\bar{x}\,|\,\bar{y})$ if and only if the channel synthesis rate R_{s} is larger than $I(\bar{X};\bar{Y})$. We point out that, when $\bar{X} \sim \mathrm{Bern}(1/2)$, the construction in [48] can be generalized to a *linear* construction[1]. Therefore, one can see that the condition of a small decoding distance on the point-to-point channel code (H, ϕ) is similar in nature to the condition imposed in the distributed channel synthesis problem, where the information bits corresponding to the output of ϕ can be seen as the index shared to the decoder in channel synthesis. Note that the encoder of the construction for the distributed channel synthesis problem would be the decoder of our point-to-point channel code, while the decoder of the channel synthesis construction is the encoder of the point-to-point code.

In short, whereas the probability of error is a measure of the *error correction* capability of the code when simulated over the symmetric channel, the decoding distance is a measure of the *shaping* capability of the decoding function. Note that polar codes under successive cancellation decoding are known to have a vanishing error probability [6] and a vanishing decoding distance [49] asymptotically over any BMS channel. In the coming chapters, we will construct coding schemes for various

[1] This holds because Theorem VII.1 in [48] only uses the *pairwise* independence of the codewords.

problems in network information theory starting from one (or more) point-to-point channel codes. The properties of the point-to-point channel codes (either error probability or decoding distance) will be instrumental in translating the performance guarantees from one communication setting to another.

2.4 The Lego-Brick Approach to Coding over Networks

Today's modern infrastructure is becoming increasingly interconnected through information networks. Emerging applications in transportation systems, power systems, smart cities, cloud computing and digital healthcare, call for exceptionally efficient coding schemes to process, store and communicate the massive amounts of network data. In practice, the common paradigm of designing coding schemes over networks is to decompose the network into separate point-to-point links, where a point-to-point channel code is used over each link. Despite the practical convenience, such an approach is known to be sub-optimal from an information theoretic perspective, even when each link is utilized at its full capacity.

Network information theory studies the fundamental limits of network communication and the optimal coding schemes that achieve those limits. At a conceptual level, this theory has been hugely successful, with several basic coding schemes that are applicable to a variety of network models, and some optimal in certain special cases. Except for a few simple use cases, however, the coding schemes developed in network information theory have barely had any impact on the design of communication systems over networks. Even basic coding schemes such as Gelfand–Pinsker coding [37], Marton coding [41] and compress-and-forward relaying [50] have not been used in practice in any meaningful manner in the forty years since their inception. The main reason behind this noticeable gap between theory and practice is that most of these coding schemes, albeit being conceptually beautiful, are not in an easily-implementable form, which is exemplified by the ubiquitous use of high-complexity coding techniques such as joint typicality encoding and decoding, and maximum likelihood decoding.

As mentioned before, this part of the dissertation is an attempt to close the aforementioned gap between theory and practice, so that all the beautiful coding schemes in network information theory (e.g., all the ones described in [28]) can be implemented in real systems to their full potential. To achieve this goal, we take a modular approach to transform the conceptual coding schemes developed in network information theory into practical implementations. More specifically, we identify basic coding schemes that are designed for one (or more) communication setting and satisfy certain properties[2], and combine them together to build a more complex coding scheme for a different communication setting. Under such a framework, we ask:

[2] The error probability and the decoding distance of a code designed for a point-to-point BMS channel are two examples of such properties.

- What are the most primitive properties that the basic coding schemes should satisfy while being versatile in building coding schemes for network communication?
- How can such coding blocks be assembled together in different network communication scenarios?
- How do the performance guarantees and the achievable rate regions translate in different communication settings?

The *Lego-brick approach* to coding over networks aims to answer these questions for various problems in network information theory. In this dissertation, we focus on the coding problems mentioned in Section 1.4.1 and Figure 1.3. As we shall see, such an approach to coding allows one to leverage commercial off-the-shelf codes that are designed for single-user symmetric channels (e.g., all the ones mentioned in Section 1.1, or even hypothetical codes to be invented in the future) to build practical coding schemes for multiuser communication.

It turns out that the only two properties of the constituent point-to-point codes which allows to translate their performance to network communication settings are the error probability and the decoding distance. Bounds on the performance of our code constructions will be derived in terms of these two properties regardless of other properties of the code. Such flexibility allows us to be tightly coupled with the most recent development in coding theory for point-to-point communication (in terms of performance), but at the same time to be completely decoupled from it (in terms of architecture).

2.5 Related Work

The approach of designing coding schemes for network information theory problems starting from simple blocks is not new in general. In [51], Wyner constructed a Slepian–Wolf code for a doubly symmetric binary source starting from a point-to-point channel code designed for the binary symmetric channel (BSC). The duality between general Slepian–Wolf problems and point-to-point channel coding problems was further explored in [52, 46], where a maximum-likelihood channel decoder was assumed. In [53], linear Slepian–Wolf codes were constructed starting from "off-the-shelf" linear channel codes designed for symmetric channels, where the exact relation of the rates and probability of error between the two problems was established. Further, a general method for constructing codes for asymmetric point-to-point channels was described in [54]; the coding scheme uses a lossless source code and a channel code designed for a symmetric channel as its constituent building blocks. Moreover, codes for the broadcast channel and the multiple access channel were constructed in [55] starting from basic coding blocks. In particular, in our recent works of [56] and [57], coding schemes for lossy source coding, asymmetric channels and channels with state were constructed starting from point-to-point channel codes designed for symmetric channels.

In a similar spirit, many attempts to design practical coding schemes for multi-terminal scenarios have closely followed the footsteps of point-to-point channel coding. Polar codes, for example, have been specialized to several problems in network information theory, including, but not limited to, the Slepian–Wolf problem [58], the lossy source coding problem [49], Gelfand–Pinsker problem [59], the multiple description coding problem [60, 44], multiple-access channels [61, 62], broadcast channels [54], interference channels [63], and relay channels [64, 65]. Sparse graph codes with logarithmic check node degrees have also been shown to achieve the optimal rates for various coding problems under maximum likelihood decoding, including the lossy source coding problem [66], the Gelfand–Pinsker problem and the Wyner–Ziv problem [67]. Alternatively, low-density generator-matrix (LDGM) codes were shown to approach the rate-distortion bound for the lossy source coding problem under variants of the low-complexity message-passing decoder [68]. Spatially coupled compound LDPC/LDGM codes were also shown to achieve optimal rates for the problems of lossy source coding, Wyner–Ziv coding and Gelfand–Pinsker coding under message-passing decoding [69, 70]. For Gaussian channels with Gaussian state that is known noncausally at the encoder (i.e., the dirty paper coding problem [71]), lattice codes have been shown to achieve capacity [72], and variants of these codes with practical decoders have been proposed in the literature (e.g., [73, 74]).

Given this introduction to the Lego-brick approach to coding and the existing literature, we are now ready to construct coding schemes for source and channel coding over networks. We start from source coding in the next chapter.

Acknowledgement

Chapter 2, in part, is a reprint with permission of the material as it appears in the paper: Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to coding for network communication,” arXiv:2211.07208, November 2022, which is in preparation to be submitted to *IEEE Transactions on Information Theory.* The dissertation author was the primary investigator and author of this paper. This work was supported in part by the Institute for Information & Communication Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-01659, 5G Open Intelligence-Defined RAN (ID-RAN) Technique based on 5G New Radio), in part by the NSERC Discovery Grant No. RGPIN-2019-05448, and in part by the NSERC Collaborative Research and Development Grant CRDPJ 543676-19.

Chapter 3

Multiterminal Source Coding: A Lego-Brick Approach

In this chapter, we deal with multiterminal source coding problems. Starting from basic coding blocks, we construct coding schemes for: Slepian–Wolf coding and lossless source coding in Section 3.1, lossy source coding in Section 3.2, Wyner–Ziv coding in Section 3.3, Berger–Tung coding in Section 3.4 and multiple description coding in Section 3.5. In Section 3.6, we show simulation results for the lossy source coding problem.

3.1 Slepian–Wolf Coding

In this section, we establish an "equivalence" between constructing a code for a linear point-to-point BMS channel and constructing a binary Slepian–Wolf code. In other words, we show that a code for any binary Slepian–Wolf problem can be designed starting from a linear code for a suitable BMS channel. Conversely, a linear code for any BMS channel can be constructed starting from a binary Slepian–Wolf code. In both cases, the optimal rate can be achieved asymptotically provided that the constituent Lego brick is rate-optimal. As a special case, the Slepian–Wolf coding scheme can be specialized to lossless source coding of a binary source. The duality between Slepian–Wolf coding and coding for a BMS channel has been previously noted in [51, 52, 46, 53]; in particular, the constructions provided in this section are equivalent to the ones in [53].

3.1.1 Problem Statement

A binary Slepian–Wolf problem $p(x, y)$ consists of a source alphabet $\mathcal{X} = \{0, 1\}$, an arbitrary side information alphabet $\mathcal{Y}$, and a joint pmf $p(x, y)$ over $\mathcal{X} \times \mathcal{Y}$ [34]. A discrete memoryless source X with side information Y generates a jointly i.i.d. random process $\{(X_i, Y_i)\}$ with

$(X_i, Y_i) \sim p(x, y)$. The goal is to represent a length-n source sequence X^n using as few bits as possible to a decoder that has access to the side information sequence Y^n and wishes to find an estimate $\widehat{X}^n$ of X^n. An (ℓ, n) code (g, ψ) for the Slepian–Wolf problem $p(x, y)$ consists of

- an index set $\mathcal{I} \subseteq \{0,1\}^n$ such that $|\mathcal{I}| = 2^\ell$,
- an encoder $g : \mathcal{X}^n \to \mathcal{I}$ that maps each source sequence x^n to an index $s^n = g(x^n)$, and
- a decoder $\psi : \mathcal{I} \times \mathcal{Y}^n \to \mathcal{X}^n$ that assigns a source estimate $\hat{x}^n = \psi(s^n, y^n)$ to each index s^n and side information sequence y^n.

The rate of the code is $R = \ell/n$. The average probability of error of the code is $\epsilon = \mathsf{P}\{\widehat{X}^n \neq X^n\}$. A rate R, $0 \leq R \leq 1$, is said to be *achievable* for the binary Slepian–Wolf coding problem if there exists a sequence of (nR, n) codes with vanishing error probability asymptotically. The classical result of Slepian and Wolf states that any rate $R > H(X \,|\, Y)$ is achievable for the Slepian–Wolf coding problem [34].

The Slepian–Wolf code is linear when the encoding function g is linear, i.e., if for any $x^n, \tilde{x}^n \in \{0,1\}^n$, we have $g(x^n \oplus \tilde{x}^n) = g(x^n) \oplus g(\tilde{x}^n)$. When the encoding function g can be defined as a matrix multiplication $g(x^n) = \begin{bmatrix} \mathbf{0} \\ Hx^n \end{bmatrix}$, where H is an $\ell \times n$ matrix, we will refer to the code as an (ℓ, n) linear Slepian–Wolf code (H, ψ).

3.1.2 Code for P2P BMS Channel → Slepian–Wolf Code

Consider a binary Slepian–Wolf problem $p(x, y)$, as defined in the previous section. We will construct a linear Slepian–Wolf code for this problem starting from a linear point-to-point channel code for a BMS channel. The BMS channel of interest is the symmetrized channel corresponding to $p(x, y)$, as defined in Section 2.2. The following lemma will be helpful to describe the Slepian–Wolf coding scheme.

Lemma 3.1.1. *Let H be a parity-check matrix for a codebook $\mathcal{C}$, and let $\widetilde{H}$ be as defined in (2.1). Define $\mathcal{S} = \{s^n \in \{0,1\}^n : s^k = 0^k\}$. Then,*

(i) For any $x^n \in \{0,1\}^n$, there exists a unique $s^n \in \mathcal{S}$ such that $x^n \oplus s^n \in \mathcal{C}$. In particular, $s^n = \widetilde{H} x^n$.

(ii) If X^n is i.i.d. Bern(1/2), *then $X^n \oplus \widetilde{H} X^n \sim$* Unif$(\mathcal{C})$.

(iii) If $C^n \sim$ Unif$(\mathcal{C})$ *and $S^n \sim$* Unif$(\mathcal{S})$ *are independent, then $C^n \oplus S^n$ is i.i.d.* Bern(1/2).

Proof. See Appendix 3.A. □

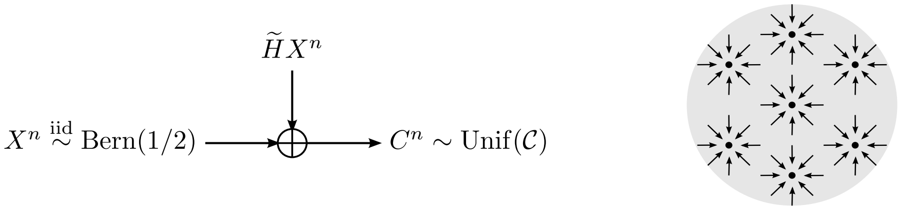

$\widetilde{H}X^n$
$X^n \overset{\text{iid}}{\sim} \text{Bern}(1/2)$
$C^n \sim \text{Unif}(\mathcal{C})$

Figure 3.1: Illustration of a shift by $\widetilde{H}X^n$ in $\{0,1\}^n$ space.

Intuitively, part (ii) of Lemma 3.1.1 gives a general way of generating a codeword uniformly at random starting from a uniformly distributed binary sequence, as illustrated in Figure 3.1. Conversely, part (iii) generates a uniformly distributed binary sequence starting from a codeword chosen uniformly at random.

The main idea of the construction of the Slepian–Wolf coding scheme is captured in the following lemma.

Lemma 3.1.2. *Let (X^n, Y^n) be i.i.d. according to $p(x,y)$, and V^n be i.i.d.* Bern(1/2) *and independent of (X^n, Y^n). Let $\bar{p}$ be the symmetrized channel corresponding to $p(x,y)$, as defined in Definition 2.2.1. Consider a parity-check matrix H for a codebook $\mathcal{C}$, and let $\widetilde{H}$ be as defined in* (2.1)*. Consider the sequences*

$$\begin{aligned} C^n &= X^n \oplus V^n \oplus \widetilde{H}X^n \oplus \widetilde{H}V^n, \\ U^n &= V^n \oplus \widetilde{H}V^n \oplus \widetilde{H}X^n. \end{aligned} \tag{3.1}$$

Then,

$$P\{C^n = c^n, U^n = u^n, Y^n = y^n\} = \frac{1}{2^k} \prod_{i=1}^{n} \bar{p}(y_i, u_i \,|\, c_i)$$

for every $c^n \in \mathcal{C}$, $u^n \in \{0,1\}^n$ and $y^n \in \mathcal{Y}^n$.

Proof. Lemma 3.1.2 can be seen as a recast of Lemmas 2 and 3 in [53]. For completion, the proof is provided in Appendix 3.B. □

Lemma 3.1.2 says that if Y^n is the output of the channel $p(y\,|\,x)$ when the channel input is X^n, then, for a uniformly distributed binary sequence V^n, the sequences (Y^n, U^n) are distributed as the outputs of the channel $\bar{p}$ when the channel input is C^n, a uniformly distributed codeword in the codebook $\mathcal{C}$, where C^n and U^n are as defined in (3.1). The relations between the different random variables is illustrated in Figure 3.2.

Now, we are ready to construct a coding scheme for the Slepian–Wolf problem $p(x,y)$. The coding scheme uses the following point-to-point channel code.

Lego Brick 3.1.1 (P2P → SW): *a (k,n) linear point-to-point channel code (H, ϕ) with codebook $\mathcal{C}$ for the symmetrized channel $\bar{p}$ corresponding to $p(x,y)$, which is defined over an input alphabet*

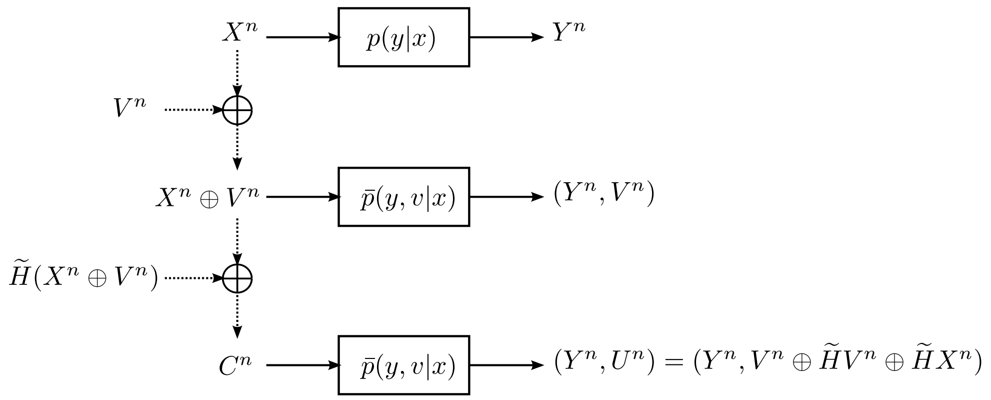

X^n
$p(y|x)$
Y^n
V^n
$X^n \oplus V^n$
$\bar{p}(y,v|x)$
(Y^n, V^n)
$\widetilde{H}(X^n \oplus V^n)$
C^n
$\bar{p}(y,v|x)$
$(Y^n, U^n) = (Y^n, V^n \oplus \widetilde{H}V^n \oplus \widetilde{H}X^n)$

Figure 3.2: The relations between the random variables (X^n, Y^n, C^n, U^n) defined in Lemma 3.1.2. Notice the similarity to Figure 2.1 when S^n in Figure 2.1 is set to $\widetilde{H}(X^n \oplus V^n)$. To recover X^n from Y^n, one can go through the path $(Y^n, U^n) \to C^n \to R^n \to X^n$. To get C^n from (Y^n, U^n), one can apply a decoder of a point-to-point channel code designed for the channel $\bar{p}$. This explains the Slepian–Wolf coding scheme shown in Fig. 3.4.

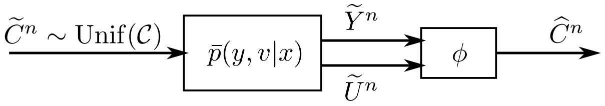


Figure 3.3: A code for the symmetric channel $\bar{p}$, defined in (3.2).

$\mathcal{X} = \{0,1\}$ and output alphabet $\mathcal{Y} \times \{0,1\}$ by

$$\bar{p}(y, v \,|\, x) = p_{X,Y}(x \oplus v, y). \tag{3.2}$$

Let ϵ be the average probability of error of the code (H, ϕ) when used over the channel $\bar{p}$.

Figure 3.3 shows the channel code (H, ϕ) when used over the channel $\bar{p}$. The average probability of error ϵ of the code can be expressed as

$$\epsilon = \mathsf{P}\{\phi(\widetilde{U}^n, \widetilde{Y}^n) \neq \widetilde{C}^n\}.$$

Figure 3.4 illustrates the block diagram of the Slepian–Wolf coding scheme that uses the point-to-point channel code (H, ϕ). The coding scheme can be described as follows.

Encoding: Upon observing the source sequence x^n, the sender transmits $s^n = \widetilde{H}x^n$, where $\widetilde{H}$ is as defined in (2.1).

Decoding: Upon observing the side information sequence y^n and receiving the index s^n, the decoder declares $\hat{x}^n = \phi(s^n \oplus v^n \oplus \widetilde{H}v^n, y^n) \oplus s^n \oplus v^n \oplus \widetilde{H}v^n$ as the source estimate, where v^n is a realization of a random dither generated independently at the decoder. Notice the similarity of this decoding method with the observations made through Fig. 3.2.

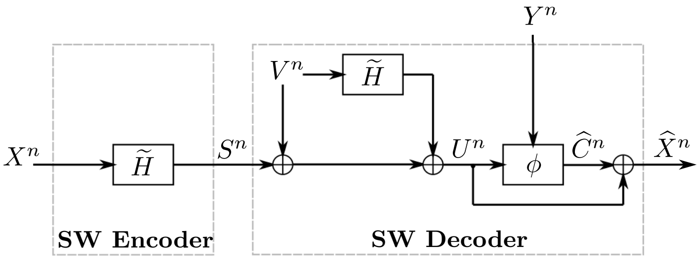


Figure 3.4: A Slepian–Wolf code starting from a point-to-point channel code.

Analysis of probability of error: We have

$$\begin{aligned}\mathsf{P}\{\widehat{X}^n \neq X^n\} &= \mathsf{P}\{\phi(V^n \oplus \widetilde{H}V^n \oplus \widetilde{H}X^n, Y^n) \oplus \widetilde{H}X^n \oplus \widetilde{H}V^n \oplus V^n \neq X^n\} \\ &= \mathsf{P}\{\phi(V^n \oplus \widetilde{H}V^n \oplus \widetilde{H}X^n, Y^n) \neq X^n \oplus V^n \oplus \widetilde{H}(X^n \oplus V^n)\} \\ &= \mathsf{P}\{\phi(U^n, Y^n) \neq C^n\} \\ &\overset{(a)}{=} \mathsf{P}\{\phi(\widetilde{U}^n, \widetilde{Y}^n) \neq \widetilde{C}^n\} \\ &= \epsilon,\end{aligned}$$

where (a) follows from Lemma 3.1.2. Note that since the probability of error averaged over V^n is ϵ, there exists a deterministic v^n sequence such that the probability of error is bounded by ϵ.

Rate: By construction, the rate of the Slepian–Wolf code is $(n-k)/n$.

Remark 3.1.1. Recall the definition of $(\bar{X}, \bar{Y})$ in Remark 2.2.1. A sequence of codes for the channel $\bar{p}$ with a vanishing error probability exists if and only if the rate is smaller than

$$I(\bar{X};\bar{Y}) = H(\bar{X}) - H(\bar{X}\,|\,\bar{Y}) = 1 - H(X \oplus U\,|\,Y, U) = 1 - H(X\,|\,Y).$$

It follows that, if the rate of the code for the channel $\bar{p}$ is $\frac{k}{n} = I(\bar{X};\bar{Y}) - \gamma$ for some $\gamma > 0$, then the rate of the Slepian–Wolf code is $\frac{n-k}{n} = H(X\,|\,Y) + \gamma$.

Conclusion: From each linear (k,n) code for the BMS channel $\bar{p}$ defined in (3.2) with average probability of error ϵ, one can construct a linear $(n-k,n)$ code for the Slepian–Wolf problem $p(x,y)$ with average probability of error ϵ.

3.1.3 Slepian–Wolf Code → Code for P2P BMS Channel

Now, we consider a BMS channel $p(y\,|\,x)$. We show that a code for this channel can be constructed starting from the following Slepian–Wolf code.

Lego Brick 3.1.2 (SW → P2P): *an $(n-k,n)$ linear Slepian–Wolf code (H,ψ) for the problem $p(x,y) = \frac{1}{2}p(y\,|\,x)$ with an average probability of error ϵ.*

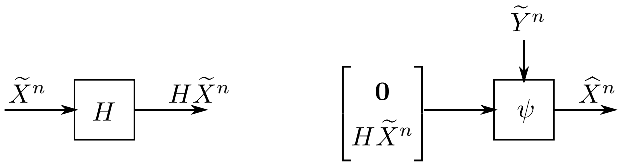


Figure 3.5: A linear Slepian–Wolf code.

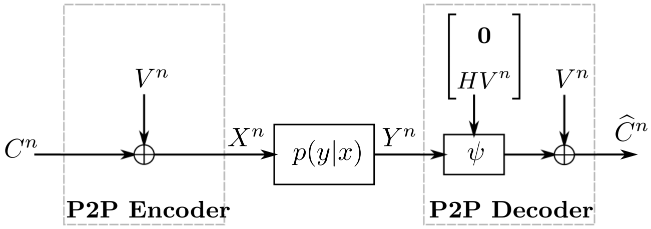


Figure 3.6: A point-to-point channel code starting from a Slepian–Wolf code.

Figure 3.5 shows the Slepian–Wolf code (H, ψ), where $(\widetilde{X}^n, \widetilde{Y}^n)$ are i.i.d. sequences distributed according to $p(x, y)$. The average probability of error ϵ of the Slepian–Wolf code can be written as

$$\epsilon = \mathsf{P}\left\{\psi\left(\begin{bmatrix}\mathbf{0}\\ H\widetilde{X}^n\end{bmatrix}, \widetilde{Y}^n\right) \neq \widetilde{X}^n\right\}.$$

To construct a code for the channel $p(y\,|\,x)$, let V^n be an i.i.d. Bern(1/2) random dither shared between the encoder and the decoder, and let $C^n \in \mathcal{C}$ represent the message to be transmitted, where $\mathcal{C}$ is the codebook corresponding to H. Figure 3.6 illustrates the block diagram of the point-to-point channel code. The coding scheme can be summarized as follows:

Encoding: To send the message $c^n \in \mathcal{C}$, the sender transmits $x^n = c^n \oplus v^n$, where v^n is a realization of a random dither shared between the encoder and the decoder.

Decoding: Upon observing y^n, the decoder declares $\hat{c}^n = \psi(\bar{H}v^n, y^n) \oplus v^n$ as the message estimate.

Analysis of probability of error: We have

$$\begin{aligned}\mathsf{P}\{\widehat{C}^n \neq C^n\} &= \mathsf{P}\left\{\psi\left(\begin{bmatrix}\mathbf{0}\\ HV^n\end{bmatrix}, Y^n\right) \oplus V^n \neq C^n\right\}\\ &\overset{(a)}{=} \mathsf{P}\left\{\psi\left(\begin{bmatrix}\mathbf{0}\\ HX^n\end{bmatrix}, Y^n\right) \neq X^n\right\}\\ &\overset{(b)}{=} \mathsf{P}\left\{\psi\left(\begin{bmatrix}\mathbf{0}\\ H\widetilde{X}^n\end{bmatrix}, \widetilde{Y}^n\right) \neq \widetilde{X}^n\right\}\\ &= \epsilon,\end{aligned}$$

where (a) follows since $HX^n = HC^n \oplus HV^n = HV^n$, and (b) follows since after dithering with the uniform V^n, (X^n, Y^n) are identically distributed as $(\widetilde{X}^n, \widetilde{Y}^n)$ in the Slepian–Wolf problem. Note that since the probability of error averaged over V^n is ϵ, there exists a deterministic v^n sequence such that the probability of error is bounded by ϵ.

Rate: Since $|\mathcal{C}| = 2^k$, the rate of the point-to-point channel code is k/n.

Remark 3.1.2. If the rate of the Slepian–Wolf code is $\frac{n-k}{n} = H(X \,|\, Y) + \gamma$ for some $\gamma > 0$, then the rate of the point-to-point channel code is

$$1 - H(X \,|\, Y) - \gamma = I\big(\mathrm{Bern}(1/2), p(y \,|\, x)\big) - \gamma,$$

where $I\big(\mathrm{Bern}(1/2), p(y \,|\, x)\big)$ is the capacity of the BMS channel $p(y \,|\, x)$.

Conclusion: From each linear $(n-k, n)$ Slepian–Wolf code for the problem $p(x, y) = \frac{1}{2}p(y \,|\, x)$ with average probability of error ϵ, one can construct a linear (k, n) code for the BMS channel $p(y \,|\, x)$ with average probability of error ϵ.

3.1.4 Specialization to Lossless Source Coding

As a special case of Slepian–Wolf coding, a lossless source code for a binary source can be implemented using a linear point-to-point channel code that is designed for a binary symmetric channel (BSC). This observation is well-understood in the literature [75]. Here, we describe an explicit construction which uses any linear off-the-shelf code designed for a BSC. The construction that we present will turn out to be useful in constructing coding schemes for other more complicated problems. The construction is very similar to the Slepian–Wolf construction in the special case of no side information.

To see this, consider a binary memoryless source that generates an i.i.d. Bern(θ) sequence X^n for some $\theta \in (0, 1/2)$. As in Slepian–Wolf coding, the goal is to represent the source sequence using as few bits as possible to a decoder that wishes to find an estimate $\widehat{X}^n$ of the sequence. The definition of a lossless source code, its rate and probability of error follow similarly as in Slepian–Wolf coding, with the exception that a lossless source decoder has no access to any side information sequence.

Recall the coding scheme presented in Section 3.1.2. When there is no side information, the symmetrized channel corresponding to $p(x)$ is a BSC($p_X(1)$). Therefore, a lossless source coding scheme for a Bern(θ) source can be constructed starting from a point-to-point channel code designed for BSC(θ), as given in the following Lego brick.

Lego Brick 3.1.3 (P2P → Lossless): *a (k, n) linear point-to-point channel code (H, ϕ) designed for* BSC(θ) *with average probability of error ϵ when used over the channel.*

Figure 3.7 shows the lossless source coding scheme, where X^n is an i.i.d. Bern(θ) source sequence, and V^n is an i.i.d. Bern(1/2) sequence generated at the decoder independently of X^n.

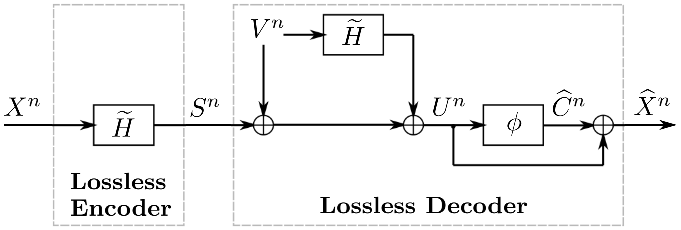


Figure 3.7: A lossless source code starting from a point-to-point channel code.

Notice the similarity of the lossless source coding scheme to the Slepian–Wolf coding scheme (Figure 3.4), except that no side information sequence is used at the decoder side. As before, the rate of the lossless source code is $(n-k)/n$, and its average probability of error is given by ϵ, which follows by specializing Lemma 3.1.2 to the case when $p(y)$ is deterministic and independent of $p(x)$.

Remark 3.1.3. If the rate of the channel code is $\frac{k}{n} = 1 - H(\theta) - \gamma$ for some $\gamma > 0$, then the rate of the lossless source code is $H(\theta) + \gamma$.

Remark 3.1.4. Conversely, a lossless source code can be used to construct a point-to-point channel code for a binary symmetric channel (BSC).

3.2 Lossy Source Coding

In this section, we construct coding schemes for the lossy source coding problem starting from simple Lego bricks. We consider two cases: the first is the case of a symmetric source, and the second corresponds to a general asymmetric source. The distinction is made because a simpler construction is possible for the former case. More specifically, in the case of a symmetric source, our coding scheme is constructed starting from a single point-to-point symmetric channel code, whereas the coding scheme for the general asymmetric source uses both a point-to-point symmetric channel code and a lossless source code. In both cases, the proposed coding scheme is rate-optimal provided that the constituent Lego bricks are rate-optimal.

3.2.1 Problem Statement

Introduced by Shannon in [35], the problem of lossy compression of a binary memoryless source generates an i.i.d. random process $\{X_i\}$ with $X_i \sim \text{Bern}(\theta)$ for some $\theta \in (0, 1/2]$. The goal is to efficiently represent a source sequence X^n when some distortion is allowed during reconstruction. More formally, an (R, n) code for the lossy source coding problem consists of

- an index set $\mathcal{I}$ such that $|\mathcal{I}| = 2^{nR}$,
- an encoder $g : \{0,1\}^n \to \mathcal{I}$ that assigns an index $m \in \mathcal{I}$ to each source sequence x^n, and

- a decoder $\psi : \mathcal{I} \to \{0,1\}^n$ that assigns an estimate $\hat{x}^n$ to each index $m \in \mathcal{I}$.

The rate of the code is R, and its expected distortion is

$$\frac{1}{n}\,\mathsf{E}\left[d_H(X^n, \widehat{X}^n)\right] = \frac{1}{n}\sum_{x^n} p(x^n) d_H\big(x^n, \psi(g(x^n))\big),$$

where $d_H(.,.)$ denotes the Hamming distance metric. A rate-distortion pair (R, D) is said to be *achievable* if there exists a sequence of (R, n) codes with

$$\limsup_{n\to\infty} \frac{1}{n}\,\mathsf{E}\left[d_H(X^n, \widehat{X}^n)\right] \leq D.$$

The *rate-distortion function* $R(D)$ is defined as the infimum of all rates R such that (R, D) is achievable.

Shannon [35] showed that the rate-distortion function for a Bern(θ) source can be expressed as

$$R(D) = \begin{cases} H(\theta) - H(D) & \text{for } 0 \leq D < \theta, \\ 0 & \text{for } D \geq \theta. \end{cases}$$

Shannon's random coding scheme assigns, for each typical source sequence x^n, a reconstruction sequence $\hat{x}^n$ that is jointly typical with x^n for some desired conditional pmf $p(\hat{x}\,|\,x)$. For the case of a binary source, the desired conditional pmf $p(\hat{x}\,|\,x)$ corresponds to the case when the "backward" channel $p(x\,|\,\hat{x})$ is a BSC(D).

3.2.2 Symmetric Source

Consider a realization of a symmetric source $X^n \overset{\text{iid}}{\sim}$ Bern(1/2), and let $D \in (0, 1/2)$ be some desired distortion level. We will construct a lossy source coding scheme for this source starting from the following point-to-point channel code. The coding scheme seeks to generate a sequence $\widehat{X}^n$ that "looks" like the output of a BSC(D) when the input is X^n.

Lego Brick 3.2.1 (P2P → Sym. Lossy): *a (k, n) linear point-to-point channel code (H, ϕ) for* BSC(D) *with a decoding distance δ.*

Recall from Section 2.3 the definition of the decoding distance of a point-to-point channel code designed for a BMS channel. The main ingredient in the lossy source coding scheme is utilizing the shaping capability of the decoding function ϕ – manifested by its decoding distance property δ – in order to generate a sequence according to the desired distribution (or "close" to it). In the simple setting of a symmetric source, this can be done by simply declaring the output of the decoding function as the source reconstruction, and the corresponding information bits as the index shared to the decoder, as depicted in Figure 3.8, where G is the generator matrix of the code (H, ϕ), and Info(.) is the function that takes as input a codeword in a linear code and outputs the corresponding information bits. The coding scheme can be summarized as follows.

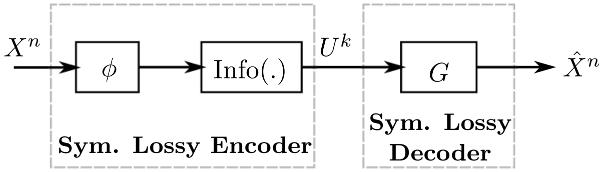


Figure 3.8: A lossy source coding scheme for a symmetric source starting from a point-to-point channel code.

Encoding: Upon observing the source sequence x^n, the encoder stores the information sequence u^k such that $\phi(x^n) = u^k G$.

Decoding: Upon observing the index u^k, the decoder declares the sequence $\hat{x}^n = u^k G$ as the source estimate.

Analysis of the average distortion: Let $q(x^n, \hat{x}^n)$ denote the distribution of $(X^n, \widehat{X}^n)$, and let $p(x^n, \hat{x}^n)$ be the desired i.i.d. distribution, i.e.,

$$p(x^n, \hat{x}^n) = \frac{1}{2^n} D^{\mathrm{wt}(x^n \oplus \hat{x}^n)} (1-D)^{n-\mathrm{wt}(x^n \oplus \hat{x}^n)}$$

Since $\widehat{X}^n = \phi(X^n)$ and $X^n \overset{\text{iid}}{\sim} \text{Bern}(1/2)$ (which is the channel output distribution of BSC(D) under the capacity-achieving input distribution), the average distortion is given by

$$\begin{aligned}\frac{1}{n}\,\mathsf{E}[d_H(X^n, \widehat{X}^n)] &= \frac{1}{n} \sum_{x^n, \hat{x}^n} q(x^n, \hat{x}^n) d_H(x^n, \hat{x}^n) \\ &= \frac{1}{n} \sum_{x^n, \hat{x}^n} p(x^n, \hat{x}^n) d_H(x^n, \hat{x}^n) + \frac{1}{n} \sum_{x^n, \hat{x}^n} \left(q(x^n, \hat{x}^n) - p(x^n, \hat{x}^n)\right) d_H(x^n, \hat{x}^n) \\ &\overset{(a)}{\leq} D + \frac{1}{2} \sum_{x^n, \hat{x}^n} |q(x^n, \hat{x}^n) - p(x^n, \hat{x}^n)| \\ &\overset{(b)}{=} D + \delta,\end{aligned}$$

where (a) holds since $p(\hat{x}^n | x^n)$ is equivalent to n independent uses of BSC(D) and the fact that $\sum_i c_i(a_i - b_i) \leq \frac{1}{2} \sum_i |a_i - b_i|$ whenever $0 \leq c_i \leq 1$ and $\sum_i a_i = \sum_i b_i$, and (b) follows by the definition of the decoding distance of the code (H, ϕ).

Rate: The rate of the coding scheme is $R = \frac{k}{n}$.

Remark 3.2.1. Following the discussion in Section 2.3, a sequence of linear point-to-point channel codes for BSC(D) with a vanishing decoding distance δ exists if (and only if[1]) the rate is larger than $1 - H(D)$.

Remark 3.2.2. Note that for this construction, it suffices to have a point-to-point channel code that satisfies

$$\left| \frac{1}{n}\,\mathsf{E}[d_H(X^n, \phi(X^n))] - D \right| \leq \delta,$$

[1] The "only if" part clearly holds by our current construction and the lossy source coding theorem.

rather than the more stringent condition of the decoding distance. Nonetheless, we show the decoding distance condition here for illustrative purposes, as it introduces the idea of using a decoding function for shaping a binary sequence, a theme that will be recurrent in several chapters of this dissertation.

Conclusion: From a point-to-point channel code for BSC(D) with decoding distance δ, one can construct an (R, n) lossy source code for a symmetric source with an expected distortion that is bounded by $D + \delta$.

3.2.3 Asymmetric Source

Now, we consider the case of a general binary memoryless source that generates an i.i.d. Bern(θ) sequence X^n for some $\theta \in (0, 1/2)$. Let $D \in (0, \theta)$ be some desired distortion level[2], and define

$$\alpha \triangleq \frac{\theta - D}{1 - 2D}.$$

Note that $\widehat{X} \sim$ Bern(α) when the conditional distribution $p(x\,|\,\hat{x})$ is BSC(D), which is the desired conditional distribution of the source given the reconstruction as inspired by Shannon's random coding scheme [35]. Let $p(x, \hat{x})$ denote the desired joint distribution between the source and the reconstruction, i.e.,

$$p(x, \hat{x}) = \alpha^{\hat{x}}(1 - \alpha)^{1-\hat{x}} D^{x \oplus \hat{x}}(1 - D)^{1-x\oplus\hat{x}}. \tag{3.3}$$

The proposed lossy source coding scheme in this general setting utilizes a point-to-point channel code and a lossless source code. At the encoder side, the point-to-point channel code is used to generate a sequence according to the desired distribution of the reconstruction (i.e., the i.i.d. Bern(α) distribution), and the lossless source code is used to compress that sequence to the decoder. Another key ingredient in the coding scheme is the assumption that the two codebooks are *nested*.

Before we describe the coding scheme, we state the following lemma, which will be useful in several constructions in this dissertation.

Lemma 3.2.1. *Let $\bar{p}(\tilde{y}, \tilde{v}\,|\,\tilde{x})$ be the symmetrized channel corresponding to a given joint distribution $p(\tilde{x}, \tilde{y})$. Let (H, ϕ) be a point-to-point channel code designed for $\bar{p}$, and let δ be its decoding distance. Let $\tilde{Y}^n$ be i.i.d. according to $p(\tilde{y})$ and $\tilde{V}^n$ be i.i.d.* Bern(1/2) *such that $\tilde{Y}^n$ and $\tilde{V}^n$ are independent, and let $\tilde{U}^n = \phi(\tilde{Y}^n, \tilde{V}^n) \oplus \tilde{V}^n$. Then,*

$$\frac{1}{2}\sum_{\tilde{u}^n, \tilde{y}^n} \left| P\{\tilde{U}^n = \tilde{u}^n, \tilde{Y}^n = \tilde{y}^n\} - \prod_{i=1}^{n} p_{\tilde{X},\tilde{Y}}(\tilde{u}_i, \tilde{y}_i) \right| \le \delta.$$

Proof. See Appendix 3.C. □

Intuitively, given an i.i.d sequence $\tilde{Y}^n$ distributed according to $p(\tilde{y})$, Lemma 3.2.1 suggests a general method of constructing a sequence $\tilde{U}^n$ such that the joint distribution of $(\tilde{Y}^n, \tilde{U}^n)$ is δ-away in total

[2] Clearly, when $D \ge \theta$, a rate-zero coding scheme is possible by deterministically outputting the all-zero sequence as the source reconstruction.

variable distance from a given joint i.i.d distribution $p(\tilde{x}, \tilde{y})$. This can be done using a point-to-point channel code with a decoding distance δ over the symmetrized channel $\bar{p}$ corresponding to $p(\tilde{x}, \tilde{y})$. This technique will be used in several constructions in the dissertation.

Now, we are ready to describe the lossy source coding scheme of a general asymmetric source. The coding scheme can be constructed from the following point-to-point channel code and lossless source code.

Lego Brick 3.2.2 (P2P → Lossy): *a (k_1, n) linear point-to-point channel code (H_1, ϕ_1) with codebook $\mathcal{C}_1$ for the channel*

$$\bar{p}(x, v \,|\, \hat{x}) = p_{X,\widehat{X}}(x, \hat{x} \oplus v). \tag{3.4}$$

Let δ denote the decoding distance of the code (H_1, ϕ_1) with respect to the channel $\bar{p}$.

Lego Brick 3.2.3 (Lossless → Lossy): *an $(n - k_2, n)$ lossless source code (H_2, ϕ_2) for a* Bern(α) *source with average probability of error ϵ. Let $\mathcal{C}_2$ be the codebook corresponding to H_2. We assume that $\mathcal{C}_2 \subseteq \mathcal{C}_1$, i.e., the two codebooks are nested.*

Remark 3.2.3. The channel $\bar{p}$ defined in (3.4) is the symmetrized channel corresponding to the joint distribution $p(x, \hat{x})$ (see Section 2.2 for a formal definition of a symmetrized channel).

Remark 3.2.4. Since $\mathcal{C}_2 \subseteq \mathcal{C}_1$, we will assume, without loss of generality, that H_1 is a submatrix of H_2, i.e., $H_2 = \begin{bmatrix} H_1 \\ Q \end{bmatrix}$ for some $(k_1 - k_2) \times n$ matrix Q.[3]

Starting from the aforementioned building blocks, Figure 3.9 shows the block diagram of the lossy source coding scheme, where V^n is an i.i.d. Bern(1/2) random dither shared between the encoder and the decoder. The lossy encoder generates the sequence $U^n = \phi_1(X^n, V^n) \oplus V^n$ which has a distribution that is δ-away in total variation distance from the i.i.d. Bern(α) distribution. This is a consequence of Lemma 3.2.1. Further, since $H_1 U^n = H_1 V^n$ and

$$H_2 U^n = \begin{bmatrix} H_1 U^n \\ Q U^n \end{bmatrix} = \begin{bmatrix} H_1 V^n \\ Q U^n \end{bmatrix},$$

the lossy decoder is able to reconstruct an estimate $\widehat{X}^n$ of the sequence U^n using only the index QU^n (since V^n is shared randomness with the decoder) and the lossless source decoder ϕ_2. The following lemma states that the joint distribution of $(X^n, \widehat{X}^n)$ is $(\delta + \epsilon)$-away in total variation distance from the desired i.i.d. $p(x, \hat{x})$ distribution.

Lemma 3.2.2. *Let $q(x^n, \hat{x}^n)$ denote the distribution of $(X^n, \widehat{X}^n)$, and let $p(x^n, \hat{x}^n)$ be the desired i.i.d. $p(x, \hat{x})$ distribution, where $p(x, \hat{x})$ is as defined in (3.3). Then,*

$$\frac{1}{2} \sum_{x^n, \hat{x}^n} |q(x^n, \hat{x}^n) - p(x^n, \hat{x}^n)| \leq \delta + \epsilon.$$

[3] Note that such a relation between H_1 and H_2 can be obtained for any pair of nested linear codes by basic row operations and column permutations.

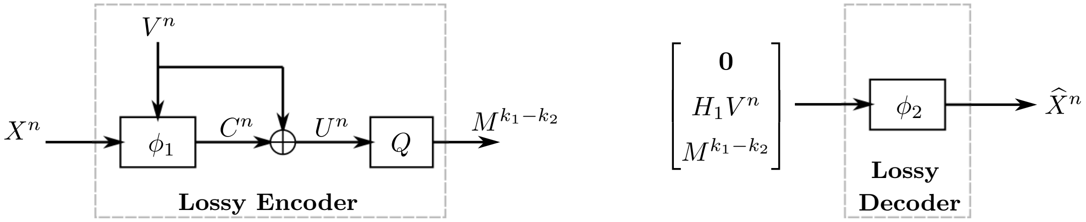


Figure 3.9: Encoder and decoder of a lossy source code for an asymmetric source starting from a point-to-point channel code and a lossless source code.

Proof. See Appendix 3.D. □

Therefore, the coding scheme can be summarized as follows.

Encoding: Upon observing the source sequence x^n, the encoder computes the sequence $u^n = \phi_1(x^n, v^n) \oplus v^n$ and transmits the index $m^{k_1-k_2} = Qu^n$, where v^n is a realization of a random dither shared with the decoder.

Decoding: Upon observing the index $m^{k_1-k_2}$, the decoder declares the sequence

$$\hat{x}^n = \phi_2 \left(\begin{bmatrix} \mathbf{0} \\ H_1 v^n \\ m^{k_1-k_2} \end{bmatrix} \right)$$

as the source estimate.

Analysis of the average distortion: The average distortion of the coding scheme can be bounded as

$$\begin{aligned}
\frac{1}{n}\mathsf{E}[d_H(X^n, \widehat{X}^n)] &= \frac{1}{n}\sum_{x^n,\hat{x}^n} q(x^n,\hat{x}^n) d_H(x^n,\hat{x}^n) \\
&= \frac{1}{n}\sum_{x^n,\hat{x}^n} p(x^n,\hat{x}^n) d_H(x^n,\hat{x}^n) + \frac{1}{n}\sum_{x^n,\hat{x}^n} \left(q(x^n,\hat{x}^n) - p(x^n,\hat{x}^n)\right) d_H(x^n,\hat{x}^n) \\
&\overset{(a)}{\le} D + \frac{1}{2}\sum_{x^n,\hat{x}^n} |q(x^n,\hat{x}^n) - p(x^n,\hat{x}^n)| \\
&\overset{(b)}{\le} D + \delta + \epsilon,
\end{aligned}$$

where (a) holds since $p(x^n \,|\, \hat{x}^n)$ is equivalent to n independent uses of BSC(D) and the fact that $\sum_i c_i(a_i - b_i) \le \frac{1}{2}\sum_i |a_i - b_i|$ whenever $0 \le c_i \le 1$ and $\sum_i a_i = \sum_i b_i$, and (b) follows by Lemma 3.2.2.

Rate: The rate of the coding scheme is $R = \frac{k_1 - k_2}{n}$.

Remark 3.2.5. A sequence of linear point-to-point channel codes for the channel $\bar{p}$ with a vanishing decoding distance δ exists if (and only if) the rate is larger than $1 - H(\widehat{X} \,|\, X)$. This follows by the discussion in Section 2.3 and the properties of a symmetrized channel (Remark 2.2.1).

Remark 3.2.6. If the rate of the point-to-point channel code is $\frac{k_1}{n} = 1 - H(\widehat{X} \,|\, X) + \gamma_1$ for some $\gamma_1 > 0$, and the rate of the lossless source code is $\frac{n-k_2}{n} = H(\alpha) + \gamma_2 = H(\widehat{X}) + \gamma_2$ for some $\gamma_2 > 0$, then the rate of the lossy source code is

$$\frac{k_1 - k_2}{n} = I(X; \widehat{X}) + \gamma_1 + \gamma_2 = H(\theta) - H(D) + \gamma_1 + \gamma_2.$$

Conclusion: Starting from a (k_1, n) linear point-to-point channel code with decoding distance δ and an $(n - k_2, n)$ lossless source code with an average probability of error ϵ, we have constructed a $(\frac{k_1-k_2}{n}, n)$ lossy source code that targets a conditional distribution $p(\hat{x} \,|\, x)$ with an average distortion that is bounded by $D + \delta + \epsilon$. Note that the point-to-point channel code should be designed for the symmetrized channel $\bar{p}$ – and not, for example, for BSC(D) – since the source is asymmetric, and, hence, the source sequence X^n is not distributed according to the channel output distribution of a BSC.

3.3 Wyner–Ziv Coding

The lossy source coding scheme presented in the previous section can be easily extended to the binary Wyner–Ziv coding problem, simply by replacing the lossless source code with a Slepian–Wolf code. Let us first review the binary Wyner–Ziv coding problem [36]. This problem consists of a source alphabet $\mathcal{X} = \{0, 1\}$, arbitrary side information alphabet $\mathcal{Y}$, a reconstruction alphabet $\widehat{\mathcal{X}} = \{0, 1\}$, and a joint pmf $p(x, y)$ over $\mathcal{X} \times \mathcal{Y}$. The source generates a jointly i.i.d. random process $\{(X_i, Y_i)\}$ with $(X_i, Y_i) \sim p(x, y)$. The goal is to efficiently represent a length-n source sequence X^n to a decoder which has access to the side information sequence Y^n and wishes to reconstruct the source sequence up to some distortion level D. The definitions of a Wyner–Ziv code, its expected distortion and achievable rates are the same as in Section 3.2.1, with the exception that the decoding function takes an additional input, namely, the side information sequence. Wyner and Ziv [36] showed that for any conditional pmf $p(\hat{x} \,|\, x)$ such that $\mathsf{E}[d(X, \widehat{X})] \le D$, any rate $R > I(X; \widehat{X} \,|\, Y)$ is achievable with a distortion level D.

In what follows, let D be some desired distortion level, and let $p(\hat{x} \,|\, x)$ be a desired conditional pmf of the reconstruction given the source such that $\mathsf{E}[d(X, \widehat{X})] \le D$, where $d(.,.)$ is the Hamming distortion metric. A code for the Wyner–Ziv coding problem can be constructed starting from the following point-to-point channel code and Slepian–Wolf code.

Lego Brick 3.3.1 (P2P $\rightarrow$ WZ): *a (k_1, n) linear point-to-point channel code (H_1, ϕ_1) with codebook $\mathcal{C}_1$ for the channel*

$$\bar{p}(x, v \,|\, \hat{x}) = p_{\widehat{X},X}(\hat{x} \oplus v, x).$$

Let δ denote the decoding distance of the code (H_1, ϕ_1) with respect to the channel $\bar{p}$.

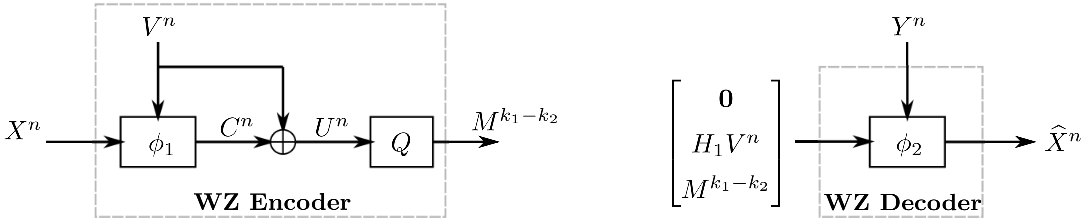


Figure 3.10: Encoder and decoder of a Wyner–Ziv coding scheme starting from a point-to-point channel code and a Slepian–Wolf code.

Lego Brick 3.3.2 (SW → WZ): *an $(n-k_2,n)$ Slepian–Wolf code (H_2,ϕ_2) for the problem*

$$p(\hat{x},y)=\sum_x p(x,y)p(\hat{x}\,|\,x),$$

with codebook $\mathcal{C}_2$ and average probability of error ϵ. We assume that the two codebooks are nested, i.e., $\mathcal{C}_2 \subseteq \mathcal{C}_1$.

Figure 3.10 shows the block diagram of the Wyner–Ziv coding scheme, where V^n is an i.i.d. Bern(1/2) random dither shared between the encoder and the decoder. The main difference in comparison to the lossy source coding scheme of Section 3.2.3 is that a Slepian–Wolf decoder is used instead of the lossless source decoder. The Slepian–Wolf decoder utilizes the available side information sequence at the decoder side. The description of the coding scheme and the analysis of its average distortion follow similarly as in the lossy source coding scheme.

Remark 3.3.1. If the rate of the point-to-point channel code is $\frac{k_1}{n}=1-H(\widehat{X}\,|\,X)+\gamma_1$ for some $\gamma_1>0$, and the rate of the Slepian–Wolf code is $\frac{n-k_2}{n}=H(\widehat{X}\,|\,Y)+\gamma_2$ for some $\gamma_2>0$, then the rate of the Wyner–Ziv code is

$$\frac{k_1-k_2}{n}=H(\widehat{X}\,|\,Y)-H(\widehat{X}\,|\,X)+\gamma_1+\gamma_2 \overset{(a)}{=} I(X;\widehat{X}\,|\,Y)+\gamma_1+\gamma_2,$$

where the equality (a) holds since Y and $\widehat{X}$ are independent given X.

Conclusion: Starting from a (k_1,n) linear point-to-point channel code with decoding distance δ and an $(n-k_2,n)$ Slepian–Wolf code with an average probability of error ϵ, we have constructed a $(\frac{k_1-k_2}{n},n)$ Wyner–Ziv code that targets a conditional distribution $p(\hat{x}\,|\,x)$ with an average distortion that is bounded by $D+\delta+\epsilon$.

3.4 Berger–Tung Coding

In this section, we describe a Berger–Tung coding scheme for distributed lossy compression starting from a lossy source code and a Wyner–Ziv code. First, we start by reviewing the problem of distributed lossy compression.

3.4.1 Problem Statement

Consider the problem of distributed lossy compression consisting of two source alphabets $\mathcal{X}_1 = \mathcal{X}_2 = \{0,1\}$, two reconstruction alphabet $\widehat{\mathcal{X}}_1 = \widehat{\mathcal{X}}_2 = \{0,1\}$, and a joint pmf $p(x_1, x_2)$ over $\mathcal{X}_1 \times \mathcal{X}_2$. Two discrete memoryless sources generate two jointly i.i.d. source sequences X_1^n and X_2^n such that $(X_{1i}, X_{2i}) \sim p(x_1, x_2)$. The goal to efficiently represent the two sequences using two separate encoders to a decoder that wishes to reconstruct the sequences with some distortion levels D_1 and D_2. More specifically, an (R_1, R_2, n) code for the distributed lossy compression problem consists of

- two index set $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $|\mathcal{M}_j| = 2^{nR_j}$, $j = 1, 2$,
- two encoders $g_j : \mathcal{X}_j^n \to \mathcal{M}_j$ for $j = 1, 2$, such that encoder g_1 assigns an index m_1 to each source sequence x_1^n, and encoder g_2 assigns an index m_2 to each source sequence x_2^n, and
- a decoder $\psi : \mathcal{M}_1 \times \mathcal{M}_2 \to \widehat{\mathcal{X}}_1^n \times \widehat{\mathcal{X}}_2^n$ that assigns a pair of estimates $(\hat{x}_1^n, \hat{x}_2^n)$ to each index pair (m_1, m_2).

A rate-distortion quadruple (R_1, R_2, D_1, D_2) is said to be achievable if there exists a sequence of (R_1, R_2, n) codes such that

$$\limsup_{n\to\infty} \frac{1}{n} \mathsf{E}[d_H(X_j^n, \widehat{X}_j^n)] \le D_j, \qquad j = 1, 2,$$

where $d_H(.,.)$ denotes the Hamming distance metric. The optimal rate-distortion region $\mathcal{R}(D_1, D_2)$ for distributed lossy compression is defined as the closure of the set of rate pairs (R_1, R_2) such that (R_1, R_2, D_1, D_2) is achievable.

The optimal rate-distortion region for the distributed compression problem is not known in general. Berger [39] and Tung [76] showed that a rate pair (R_1, R_2) is achievable for the distributed lossy compression problem with distortion pair (D_1, D_2) if

$$\begin{aligned} R_1 &> I(X_1; \widehat{X}_1 \,|\, \widehat{X}_2), \\ R_2 &> I(X_2; \widehat{X}_2 \,|\, \widehat{X}_1), \\ R_1 + R_2 &> I(X_1, X_2; \widehat{X}_1, \widehat{X}_2) \end{aligned} \tag{3.5}$$

for some conditional pmf $p(\hat{x}_1 \,|\, x_1) p(\hat{x}_2 \,|\, x_2)$ such that $\mathsf{E}[d_H(X, \widehat{X}_j)] \le D_j$, $j = 1, 2$.

3.4.2 Coding Scheme

Consider two conditional pmf's $p(\hat{x}_1 \,|\, x_1)$ and $p(\hat{x}_2 \,|\, x_2)$ such that $\mathsf{E}[d_H(X, \widehat{X}_j)] \le D_j$, for $j = 1, 2$. This completely specifies the source-reconstruction joint distribution as

$$p(x_1, x_2, \hat{x}_1, \hat{x}_2) = p(x_1, x_2) p(\hat{x}_1 \,|\, x_1) p(\hat{x}_2 \,|\, x_2).$$

The coding scheme for distributed lossy compression uses a lossy source code and a Wyner–Ziv code that target the desired conditional pmf's, as described in the following Lego-brick definitions.

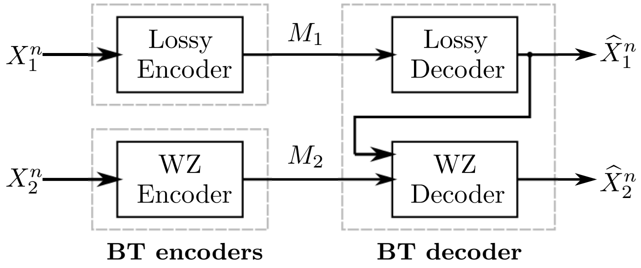


Figure 3.11: Coding scheme for distributed lossy compression using a lossy source code and a Wyner–Ziv code.

Lego Brick 3.4.1 (**Lossy** → **BT**): *an (R_1, n) lossy source code (g_1, ψ_1) for a $p(x_1)$-source that targets a conditional distribution $p(\hat{x}_1 \,|\, x_1)$ s.t. for $X_1^n \overset{\text{iid}}{\sim} p(x_1)$ and $\widehat{X}_1^n = \psi_1\big(g_1(X_1^n)\big)$, we have*

$$\frac{1}{2} \sum_{x_1^n, \hat{x}_1^n} \left| P\{X_1^n = x_1^n, \widehat{X}_1^n = \hat{x}_1^n\} - \prod_{i=1}^{n} p(x_{1i}) p(\hat{x}_{1i} \,|\, x_{1i}) \right| \le \delta_1 \tag{3.6}$$

for some $\delta_1 > 0$.

Lego Brick 3.4.2 (**WZ** → **BT**): *an (R_2, n) Wyner–Ziv code (g_2, ψ_2) for a $p(x_2, \hat{x}_1)$-source that targets a conditional distribution $p(\hat{x}_2 \,|\, x_2)$ s.t. for $(X_2^n, \widehat{X}_1^n) \overset{\text{iid}}{\sim} p(x_2, \hat{x}_1)$ and $\widehat{X}_2^n = \psi_2\big(g_2(X_2^n), \widehat{X}_1^n\big)$, we have*

$$\frac{1}{2} \sum_{x_2^n, \hat{x}_2^n} \left| P\{X_2^n = x_2^n, \widehat{X}_2^n = \hat{x}_2^n\} - \prod_{i=1}^{n} p(x_{2i}) p(\hat{x}_{2i} \,|\, x_{2i}) \right| \le \delta_2 \tag{3.7}$$

for some $\delta_2 > 0$.

Remark 3.4.1. Note that a lossy source code satisfying condition (3.6) can be constructed starting from a point-to-point channel code and a lossless source code, as described in Section 3.2.3. Also, a Wyner–Ziv code satisfying condition (3.7) can be constructed starting from a point-to-point channel code and a Slepian–Wolf code, as described in Section 3.3.

Figure 3.11 shows the block diagram of a distributed lossy compression code that uses the aforementioned Lego bricks. The coding scheme can be summarized as follows.

Encoding: Upon observing the source sequence x_1^n, the first encoder transmits the index $m_1 = g_1(m_1)$ to the decoder. Similarly, upon observing the source sequence x_2^n, the second encoder transmits the index $m_2 = g_2(m_2)$ to the decoder.

Decoding: Upon observing the index pair (m_1, m_2), the decoder declares the sequence $\hat{x}_1^n = \psi_1(m_1)$ as the source estimate of the first source sequence, and the sequence $\hat{x}_2^n = \psi_2(m_2, \hat{x}_1^n)$ as the source estimate of the second source sequence.

Analysis of the average distortion: Similar to the distortion analysis in Section 3.2.3, it can be shown using conditions (3.6) and (3.7) that the average distortions of the Berger–Tung coding scheme can be bounded as

$$\frac{1}{n}\,\mathsf{E}[d_H(X_1^n, \widehat{X}_1^n)] \leq D_1 + \delta_1,$$
$$\frac{1}{n}\,\mathsf{E}[d_H(X_2^n, \widehat{X}_2^n)] \leq D_2 + \delta_1 + \delta_2.$$

Rate: The coding scheme attains the rate pair (R_1, R_2).

Remark 3.4.2. If the rate of the lossy source code is $R_1 = I(X_1; \widehat{X}_1) + \gamma_1$ for some $\gamma_1 > 0$, and the rate of the Wyner–Ziv code is $R_2 = I(X_2; \widehat{X}_2 \,|\, \widehat{X}_1) + \gamma_2$ for some $\gamma_2 > 0$, then the coding scheme can achieve the rate pair

$$(R_1, R_2) = \Big(I(X_1; \widehat{X}_1) + \gamma_1, I(X_2; \widehat{X}_2 \,|\, \widehat{X}_1) + \gamma_2\Big).$$

Note that $\Big(I(X_1; \widehat{X}_1), I(X_2; \widehat{X}_2 \,|\, \widehat{X}_1)\Big)$ is a corner point of the rate-distortion region given in (3.5). By changing the order of decoding and switching the roles of the lossy source code and the Wyner–Ziv code, another corner point can be approached.

Remark 3.4.3. A similar coding scheme can be constructed for a distributed compression problem with L sources using one lossy source code and $L-1$ Wyner–Ziv codes, where the decoder successively decodes the source sequences.

Conclusion: Starting from an (R_1, n) lossy source code and an (R_2, n) Wyner–Ziv code, we constructed an (R_1, R_2, n) Berger–Tung code.

3.5 Multiple Description Coding

In this part, we describe a coding scheme for the multiple description coding problem starting from two lossy source codes and one Wyner–Ziv code. For illustrative purposes, each of the lossy source codes and Wyner–Ziv code will be described using their respective constituent codes, as described in previous sections. Therefore, we will construct a multiple description coding scheme starting from three point-to-point channel codes, two lossless source codes and a Slepian–Wolf code. Provided that these constituent codes are rate-optimal, the coding scheme can achieve a corner point of the El Gamal–Cover rate-distortion region for this problem. First, we start by describing the multiple description coding problem.

3.5.1 Problem Statement

Initially formulated by Gersho and Witsenhausen, and further studied by Wolf, Wyner, Ziv, El Gamal, and Cover [38, 77, 78, 79], the multiple description coding problem studies the rates at which a discrete memoryless source can be represented to multiple decoders through multiple descriptions. In particular, we consider a binary source sequence $X^n \overset{\text{iid}}{\sim} \text{Bern}(\theta)$ to be encoded through two descriptions such that each description by itself can be used to reconstruct the source with some distortions D_1 and D_2, and the two descriptions together can be used to reconstruct the source with a lower distortion D_0, as depicted in Figure 3.12. The goal is to characterize the optimal tradeoff between the description rate pair (R_1, R_2) and the distortion triple (D_0, D_1, D_2). An (R_1, R_2, n) *multiple description code* consists of

- two index sets $\mathcal{I}_1$ and $\mathcal{I}_2$ such that $|\mathcal{I}_j| = 2^{nR_j}$, $j = 1, 2$,
- an encoder $g : \{0,1\}^n \to \mathcal{I}_1 \times \mathcal{I}_2$ that assigns two indices m_1 and m_2 to each source sequence x^n, and
- three decoders ψ_j, $j = 0, 1, 2$, such that decoder ψ_1 assigns an estimate $\hat{x}_1^n$ to each index $m_1 \in \mathcal{I}_1$, decoder ψ_2 assigns an estimate $\hat{x}_2^n$ to each index $m_2 \in \mathcal{I}_2$, and decoder ψ_0 assigns an estimate $\hat{x}_0^n$ to each index pair $(m_1, m_2) \in \mathcal{I}_1 \times \mathcal{I}_2$.

A rate-distortion quintuple $(R_1, R_2, D_0, D_1, D_2)$ is said to be *achievable* if there exists a sequence of (R_1, R_2, n) codes such that

$$\limsup_{n\to\infty} \frac{1}{n} \mathsf{E}[d_H(X^n, \widehat{X}_j^n)] \le D_j \qquad j = 0, 1, 2,$$

where $d_H(.,.)$ denotes the Hamming distance metric. The optimal rate-distortion region $\mathcal{R}(D_0, D_1, D_2)$ is defined as the closure of the set of rate pairs (R_1, R_2) such that $(R_1, R_2, D_0, D_1, D_2)$ is achievable.

The optimal rate–distortion region for the multiple description coding problem is not known in general. A number of random-coding-based achievability results have been proposed by El Gamal and Cover [79], Chen, Tian, Berger, and Hemami [80], Berger and Zhang [81], among others. In particular, a rate pair (R_1, R_2) is achievable by El Gamal and Cover's coding scheme [79] with distortions (D_0, D_1, D_2) if

$$\begin{aligned} R_1 &> I(X; \widehat{X}_1) \\ R_2 &> I(X; \widehat{X}_2) \\ R_1 + R_2 &> I(X; \widehat{X}_0, \widehat{X}_1, \widehat{X}_2) + I(\widehat{X}_1; \widehat{X}_2) \end{aligned} \tag{3.8}$$

for some conditional pmf $p(\hat{x}_0, \hat{x}_1, \hat{x}_2 \,|\, x)$ such that $\mathsf{E}[d_H(X, \widehat{X}_j)] \le D_j$, $j = 0, 1, 2$. The coding scheme that we present next targets a corner point in El Gamal and Cover's rate region.

3.5.2 Coding Scheme

Inspired by El Gamal–Cover characterization of an achievable rate region, let $p(\hat{x}_0, \hat{x}_1, \hat{x}_2 \,|\, x)$ be some desired conditional pmf that satisfies $\mathsf{E}[d_H(X, \widehat{X}_j)] \le D_j$, for each $j = 0, 1, 2$. Therefore, the source-reconstruction joint distribution can be written as

$$p(x, \hat{x}_0, \hat{x}_1, \hat{x}_2) = p(x)p(\hat{x}_0, \hat{x}_1, \hat{x}_2 \,|\, x).$$

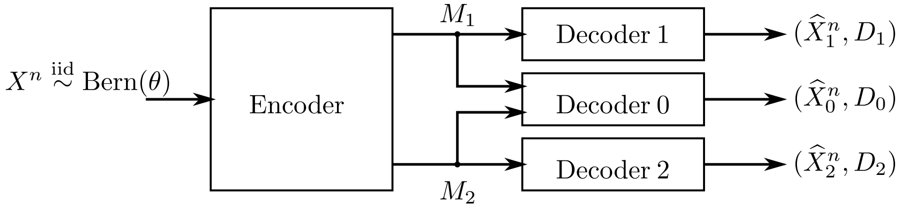


Figure 3.12: A multiple description code.

As in the construction of the lossy source coding scheme in Section 3.2.3, the main idea at the encoder side will be to generate sequences according to the desired distributions of the reconstructions, and then use lossless source coding to convey these sequences to the decoders. Hence, the coding scheme for the multiple description coding problem can be constructed starting from the following Lego bricks.

Lego Brick 3.5.1 (P2P → MDC): *a (k_{11}, n) linear point-to-point channel code (H_{11}, ϕ_{11}) with codebook $\mathcal{C}_{11}$ for the channel*

$$\bar{p}_1(x, v \,|\, \hat{x}_1) = p_{X,\widehat{X}_1}(x, \hat{x}_1 \oplus v). \tag{3.9}$$

Let δ_1 denote the decoding distance of the code (H_{11}, ϕ_{11}) with respect to the channel $\bar{p}_1$.

Lego Brick 3.5.2 (Lossless → MDC): *an $(n - k_{12}, n)$ lossless source code (H_{12}, ϕ_{12}) for a* Bern$(p_{\widehat{X}_1}(1))$ *source with codebook $\mathcal{C}_{12}$ and average probability of error ϵ_1. We further assume that $\mathcal{C}_{12} \subseteq \mathcal{C}_{11}$.*

Lego Brick 3.5.3 (P2P → MDC): *a (k_{21}, n) linear point-to-point channel code (H_{21}, ϕ_{21}) with codebook $\mathcal{C}_{21}$ for the channel*

$$\bar{p}_2(x, \hat{x}_1, v \,|\, \hat{x}_2) = p_{X,\widehat{X}_1,\widehat{X}_2}(x, \hat{x}_1, \hat{x}_2 \oplus v). \tag{3.10}$$

Let δ_2 denote the decoding distance of the code (H_{21}, ϕ_{21}) with respect to the channel $\bar{p}_2$.

Lego Brick 3.5.4 (Lossless → MDC): *an $(n - k_{22}, n)$ lossless source code (H_{22}, ϕ_{22}) for a* Bern$(p_{\widehat{X}_2}(1))$ *source with codebook $\mathcal{C}_{22}$ and average probability of error ϵ_2. We further assume that $\mathcal{C}_{22} \subseteq \mathcal{C}_{21}$.*

Lego Brick 3.5.5 (P2P → MDC): *a (k_{01}, n) linear point-to-point channel code (H_{01}, ϕ_{31}) with codebook $\mathcal{C}_{01}$ for the channel*

$$\bar{p}_0(x, \hat{x}_1, \hat{x}_2, v \,|\, \hat{x}_0) = p_{X,\widehat{X}_1,\widehat{X}_2,\widehat{X}_0}(x, \hat{x}_1, \hat{x}_2, \hat{x}_0 \oplus v). \tag{3.11}$$

Let δ_0 denote the decoding distance of the code (H_{01}, ϕ_{01}) with respect to the channel $\bar{p}_0$.

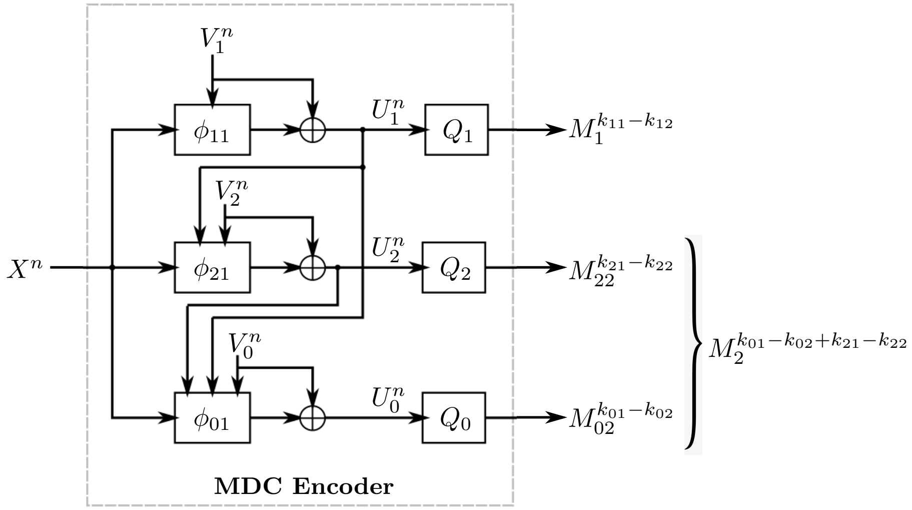


Figure 3.13: Encoder of a multiple description code using three point-to-point channel codes.

Lego Brick 3.5.6 (SW → MDC): *an $(n - k_{02}, n)$ Slepian–Wolf code (H_{02}, ϕ_{02}) for the problem $p\big(\hat{x}_0, (\hat{x}_1, \hat{x}_2)\big)$ with codebook $\mathcal{C}_{02}$ and average probability of error ϵ_0. We further assume that $\mathcal{C}_{02} \subseteq \mathcal{C}_{01}$.*

Remark 3.5.1. Due to nestedness, we will assume, without loss of generality, that H_{j1} is a submatrix of H_{j2} for each $j = 0, 1, 2$, i.e., $H_{j2} = \begin{bmatrix} H_{j1} \\ Q_j \end{bmatrix}$ for some $(k_{j1} - k_{j2}) \times n$ matrix Q_j.

Remark 3.5.2. The point-to-point channels $\bar{p}_1$, $\bar{p}_2$ and $\bar{p}_0$ are the symmetrized channels corresponding to the joint distributions $p\big(\hat{x}_1, x\big)$, $p\big(\hat{x}_2, (\hat{x}_1, x)\big)$ and $p\big(\hat{x}_0, (\hat{x}_1, \hat{x}_2, x)\big)$, respectively.

Figure 3.13 and Figure 3.14 show the block diagrams of the encoder and decoder of the multiple description code, respectively, where (V_0^n, V_1^n, V_2^n) are i.i.d. Bern(1/2) random dithers such that V_1^n is shared with decoder 1, V_2^n is shared with decoder 2, and all three random dithers are shared with decoder 0. Similar to lossy source coding, the basic idea of the coding scheme is to generate three sequences (U_0^n, U_1^n, U_2^n) whose distribution is "close" to the i.i.d. $p(\hat{x}_0, \hat{x}_1, \hat{x}_2)$ distribution, and then lossless source compression to recover estimates of the sequences at the decoders. The construction of the sequences (U_0^n, U_1^n, U_2^n) is done successively at the encoder side (Figure 3.13). That is, first, the sequence U_1^n is generated using the decoding function ϕ_{11}. Then, U_1^n is inputted to the decoding function ϕ_{21} to construct the sequence U_2^n. Intuitively, this step attempts to generate U_2^n according to the conditional distribution $p(\hat{x}_2 \,|\, x, \hat{x}_1)$. Similarly, (U_1^n, U_2^n) are inputted to the decoding function ϕ_{01} to construct the sequence U_0^n according to the conditional distribution $p(\hat{x}_0 \,|\, x, \hat{x}_1, \hat{x}_2)$. By repeated applications of Lemma 3.2.1 and the definition of decoding distance, it

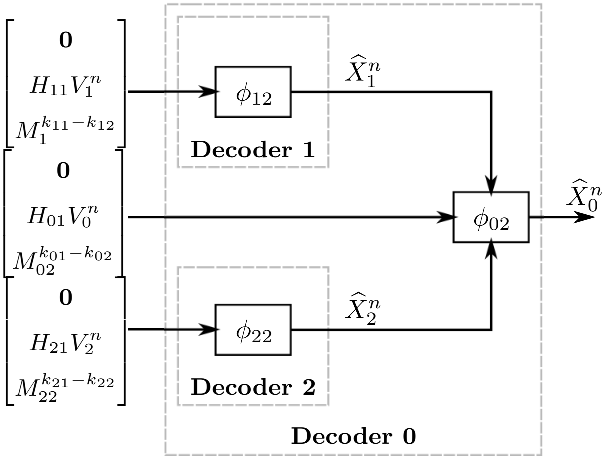


Figure 3.14: Decoder of a multiple description code using two lossless source decoders and a Slepian–Wolf decoder.

follows that

$$\frac{1}{2}\sum_{u_0^n,u_1^n,u_2^n}\left|\mathsf{P}\{U_0^n = u_0^n, U_1^n = u_1^n, U_2^n = u_2^n\} - \prod_{i=1}^{n} p_{\widehat{X}_0,\widehat{X}_1,\widehat{X}_2}(u_{0i}, u_{1i}, u_{2i})\right| \le \delta_0 + \delta_1 + \delta_2.$$

Moreover, for each $j = 0, 1, 2$, we have

$$H_{j2}U_j^n = \begin{bmatrix} H_{j1} \\ Q_j \end{bmatrix} U_j^n = \begin{bmatrix} H_{j1}V_j^n \\ Q_jU_j^n \end{bmatrix},$$

where $Q_jU_j^n$, $j = 0, 1, 2$, are indices transmitted to the decoders. Therefore, the decoder can recover the estimates of the source sequence knowing the shared indices and random dithers. The coding scheme can be summarized as follows.

Encoding: Upon observing the source sequence x^n, the encoder computes the sequence $u_1^n = \phi_{11}(x^n, v_1^n) \oplus v_1^n$ and transmits the index $m_1^{k_{11}-k_{12}} = Q_1u_1^n$ to decoder 1 and decoder 0, where v_1^n is a realization of a random dither shared with the decoders. The encoder then computes the sequences $u_2^n = \phi_{21}(x^n, u_1^n, v_2^n) \oplus v_2^n$ and $u_0^n = \phi_{01}(x^n, u_1^n, u_2^n, v_0^n) \oplus v_2^n$, and transmits the index pair

$$(m_{22}^{k_{21}-k_{22}}, m_{02}^{k_{01}-k_{02}}) = (Q_2u_2^n, Q_0u_0^n)$$

to decoder 2 and decoder 0, where v_2^n is a random dither shared with decoder 2 and decoder 0, and v_0^n is a random dither shared with the decoder 0.

Decoding: Upon observing the index $m_1^{k_{11}-k_{12}}$, decoder 1 declares the sequence

$$\hat{x}_1^n = \phi_{12}\left(\begin{bmatrix} \mathbf{0} \\ H_{11}v_1^n \\ m_1^{k_{11}-k_{12}} \end{bmatrix}\right) \tag{3.12}$$

as the source estimate. Upon observing the index $m_{22}^{k_{21}-k_{22}}$, decoder 2 declares the sequence

$$\hat{x}_2^n = \phi_{22}\left(\begin{bmatrix} \mathbf{0} \\ H_{21} v_2^n \\ m_{22}^{k_{21}-k_{22}} \end{bmatrix}\right) \tag{3.13}$$

as the source estimate. And upon observing the index triplet $(m_1^{k_{11}-k_{12}}, m_{22}^{k_{21}-k_{22}}, m_{02}^{k_{01}-k_{02}})$, decoder 0 first computes the sequences $\hat{x}_1^n$ and $\hat{x}_2^n$ as in (3.12) and (3.13), and declares the sequence

$$\hat{x}_0^n = \phi_{02}\left(\begin{bmatrix} \mathbf{0} \\ H_{01} v_0^n \\ m_{02}^{k_{01}-k_{02}} \end{bmatrix}, \hat{x}_1^n, \hat{x}_2^n\right)$$

as the source estimate (recall that $\phi_{02}(.)$ is a Slepian–Wolf decoder with side information sequences $\hat{x}_1^n$ and $\hat{x}_2^n$).

Analysis of the average distortion: Similar to the analysis in lossy source coding, it can be shown that the average distortions can be bounded as

$$\begin{aligned} \frac{1}{n}\mathsf{E}[d_H(X^n, \widehat{X}_1^n)] &\leq D_1 + \delta_1 + \epsilon_1 \\ \frac{1}{n}\mathsf{E}[d_H(X^n, \widehat{X}_2^n)] &\leq D_2 + \delta_1 + \delta_2 + \epsilon_2 \\ \frac{1}{n}\mathsf{E}[d_H(X^n, \widehat{X}_0^n)] &\leq D_0 + \delta_0 + \delta_1 + \delta_2 + \epsilon_0 + \epsilon_1 + \epsilon_2 \end{aligned}$$

Rate: The coding scheme attains the rate pair

$$(R_1, R_2) = \left(\frac{k_{11} - k_{12}}{n}, \frac{k_{21} - k_{22} + k_{01} - k_{02}}{n}\right).$$

Remark 3.5.3. Following the discussion in Section 2.3 and the properties of a symmetrized channel (Remark 2.2.1), it follows that if the rates of point-to-point channel codes are

$$\begin{aligned} \frac{k_{11}}{n} &= 1 - H(\widehat{X}_1 \,|\, X) + \gamma_{11}, \\ \frac{k_{21}}{n} &= 1 - H(\widehat{X}_2 \,|\, X, \widehat{X}_1) + \gamma_{21}, \\ \frac{k_{01}}{n} &= 1 - H(\widehat{X}_0 \,|\, X, \widehat{X}_1, \widehat{X}_2) + \gamma_{01}, \end{aligned}$$

for some γ_{11}, γ_{21}, $\gamma_{01} > 0$, and the rates of the two lossless source codes and Slepian–Wolf code are, respectively,

$$\begin{aligned} \frac{n - k_{12}}{n} &= H(\widehat{X}_1) + \gamma_{12}, \\ \frac{n - k_{22}}{n} &= H(\widehat{X}_2) + \gamma_{22}, \\ \frac{n - k_{02}}{n} &= H(\widehat{X}_0 \,|\, \widehat{X}_1, \widehat{X}_2) + \gamma_{02}, \end{aligned}$$

for some γ_{12}, γ_{22}, $\gamma_{02} > 0$, then the multiple description coding scheme attains the rate pair

$$\begin{aligned}(R_1, R_2) &= \left(\frac{k_{11} - k_{12}}{n}, \frac{k_{21} - k_{22} + k_{01} - k_{02}}{n}\right) \\ &= \big(I(X; \widehat{X}_1) + \gamma_{11} + \gamma_{12},\ I(X, \widehat{X}_1; \widehat{X}_2) + I(X; \widehat{X}_0 \,|\, \widehat{X}_1, \widehat{X}_2) + \gamma_{21} + \gamma_{22} + \gamma_{01} + \gamma_{02}\big).\end{aligned}$$

Note that the rate pair $\big(I(X; \widehat{X}_1), I(X, \widehat{X}_1; \widehat{X}_2) + I(X; \widehat{X}_0 \,|\, \widehat{X}_1, \widehat{X}_2)\big)$ is a corner point of the El Gamal–Cover rate-distortion region given in (3.8). By reversing the order of generating the sequences U_1^n and U_2^n at the encoder side, another corner point can be achieved.

Conclusion: Starting from three point-to-point channel codes, two lossless source codes and a Slepian–Wolf code, a code for the multiple description coding problem can be constructed.

3.6 Simulation Results: Lossy Source Coding

In this section, we simulate the lossy source coding scheme of Figure 3.9 for a Bern(0.3) source (i.e., $\theta = 0.3$) using polar codes with successive cancellation decoding as the constituent point-to-point channel codes. The lossless source decoder used in the construction can be implemented using a polar code designed for a binary symmetric channel, as described in Section 3.1.4. To construct the polar codes (i.e., identify the information sets), we use Arıkan's method of sorting upper bounds on the Bhattacharyya parameters of the synthetic polar bit-channels [6]. Since our coding scheme requires that the two codes are nested, the information set corresponding to the polar code (H_2, ϕ_2) is chosen to be a subset of that of the code (H_1, ϕ_1).

We first consider the encoder of Figure 3.9. We would like to see if the decoder ϕ_1 correctly shapes the sequence U^n according to the desired distribution. To this end, let D_{enc} and α_{enc} denote respectively the distortion level and the bias of the sequence U^n at the encoder side, i.e.,

$$\begin{aligned}D_{\text{enc}} &= \frac{1}{n}\, \mathsf{E}[d_H(U^n, X^n)], \\ \alpha_{\text{enc}} &= \frac{1}{n}\, \mathsf{E}[\text{wt}(U^n)],\end{aligned}$$

where $d_H(.,.)$ denotes the Hamming distance metric and wt(.) denotes the Hamming weight. Figure 3.15 shows the plot of the achieved distortion level and bias at the encoder side for a block length $n = 1024$. For comparison, the desired distortion and bias are also plotted. Note that the desired bias α corresponds to the mapping $D \mapsto \frac{\theta - D}{1 - 2D}$. At each distortion level D, the rate of the polar code (H_1, ϕ_1) is chosen to be close to the theoretical limit, i.e., we take $\frac{k_1}{n} \approx 1 - H(\widehat{X} \,|\, X)$. The results demonstrate that the achieved distortion and bias at the encoder side follow closely the desired design values. This implies that polar codes indeed have good shaping properties, even at finite block length.

Next, the entire lossy source coding scheme of Figure 3.9 is simulated. The rate-distortion tradeoff for the coding scheme is shown in Figure 3.16 for different block lengths $n = 256$, $n = 1024$,

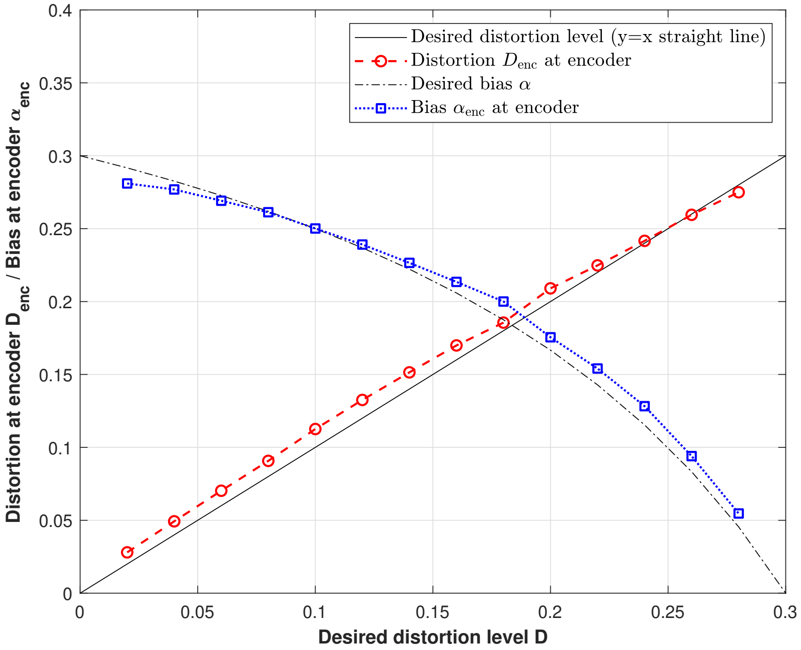


Figure 3.15: Distortion level and bias of the sequence U^n at the encoder side assuming a Bern(0.3) source and a polar code of block length $n = 1024$.

and $n = 4096$. For good error performance of the lossless source decoder, the rate $\frac{k_2}{n}$ should be chosen with a small gap to its theoretical limit, i.e., we take $\frac{k_2}{n} = 1 - H(\widehat{X}) - \gamma$ for some small "back-off" parameter $\gamma > 0$. In our simulations, we used $\gamma = 1/8$. Each simulation point shown in Figure 3.16 corresponds to a chosen rate pair $(\frac{k_1}{n}, \frac{k_2}{n})$ of the constituent channel codes, where the rate of the coding scheme is $\frac{k_1 - k_2}{n}$. For reference, the rate-distortion function is also shown. Clearly, the practical performance approaches the theoretical limit for increasing block lengths. The simulation results demonstrate that off-the-shelf codes can be leveraged in the construction of practical lossy source coding schemes. For more details about the simulation setup, our code is available on GitHub [82].

Acknowledgement

Chapter 3, in part, is a reprint with permission of the material as it appears in the papers: Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, "A Lego-brick approach to coding for network communication," arXiv:2211.07208, November 2022, which is in preparation to be submitted to *IEEE Transactions on Information Theory*, and Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, "A Lego-brick approach to lossy source coding," in *2022 17th Canadian Workshop on Information Theory (CWIT)*, pp. 45–50, 2022. The dissertation author was the primary investigator and author of these papers. This work was supported in part by

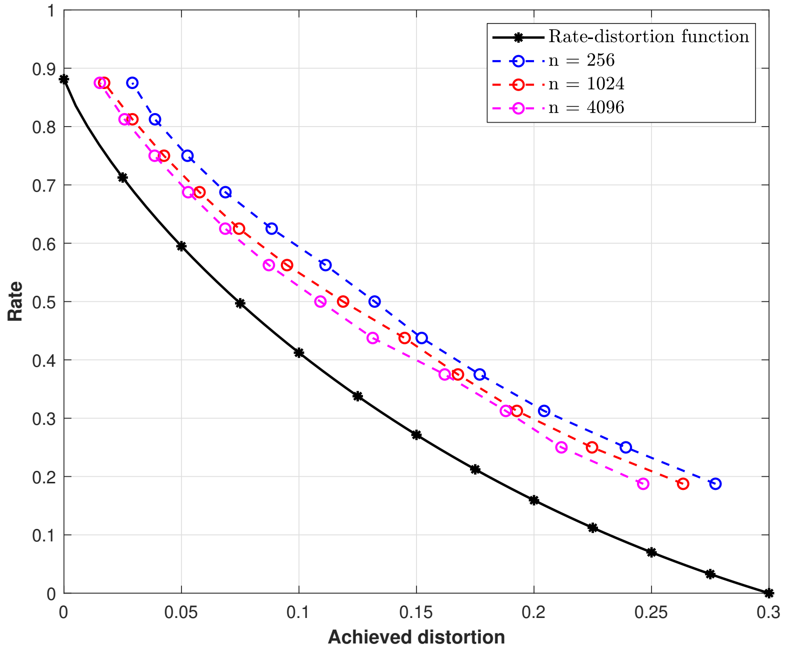


Figure 3.16: Rate-distortion tradeoff achieved by the lossy source coding scheme for a Bern(0.3) source using polar codes of different block lengths.

the Institute for Information & Communication Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-01659, 5G Open Intelligence-Defined RAN (ID-RAN) Technique based on 5G New Radio), in part by the NSERC Discovery Grant No. RGPIN-2019-05448, and in part by the NSERC Collaborative Research and Development Grant CRDPJ 543676-19.

Appendix

3.A Proof of Lemma 3.1.1

(i) Let $x^n \in \{0,1\}^n$, and let $\widetilde{H}$ be as defined in (2.1). Then $s^n \triangleq \widetilde{H}x^n \in \mathcal{S}$, and

$$H(x^n \oplus s^n) = Hx^n \oplus \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ B^{-1}A & I \end{bmatrix} x^n = Hx^n \oplus Hx^n = 0,$$

where $H = \begin{bmatrix} A & B \end{bmatrix}$ for some nonsingular matrix B. Therefore, $x^n \oplus s^n \in \mathcal{C}$. Now, assume there exists $s_1^n, s_2^n \in \mathcal{S}$, $s_1^n \neq s_2^n$, such that $x^n \oplus s_1^n \in \mathcal{C}$ and $x^n \oplus s_2^n \in \mathcal{C}$. Thus, $Hs_1^n = Hs_2^n$, which implies that $Bs_{1,k+1}^n = Bs_{2,k+1}^n$, and, therefore, $s_1^n = s_2^n$. A contradiction.

(ii) Let $C^n = X^n \oplus \widetilde{H}X^n$. For any $c^n \in \mathcal{C}$, we have

$$\begin{aligned}
\mathsf{P}\{C^n = c^n\} &= \sum_{s^n \in \mathcal{S}} \mathsf{P}\{X^n \oplus \widetilde{H}X^n = c^n, \widetilde{H}X^n = s^n\} \\
&= \sum_{s^n \in \mathcal{S}} \mathsf{P}\{X^n = c^n \oplus s^n\}\, \mathsf{P}\{\widetilde{H}X^n = s^n \,|\, X^n = c^n \oplus s^n\} \\
&\overset{(a)}{=} \sum_{s^n \in \mathcal{S}} \frac{1}{2^n} \\
&= \frac{1}{2^k},
\end{aligned}$$

where (a) follows since for any $c^n \in \mathcal{C}$ and $s^n \in \mathcal{S}$,

$$\widetilde{H}(c^n \oplus s^n) = 0^n \oplus \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ B^{-1}A & I \end{bmatrix} \begin{bmatrix} 0^k \\ s_{k+1}^n \end{bmatrix} = s^n.$$

(iii) Let $V^n = C^n \oplus S^n$. For any $v^n \in \{0,1\}^n$, we have

$$\begin{aligned}
\mathsf{P}\{V^n = v^n\} &= \sum_{s^n \in \mathcal{S}} \mathsf{P}\{V^n = v^n, S^n = s^n\} \\
&= \sum_{s^n \in \mathcal{S}} \mathsf{P}\{C^n = v^n \oplus s^n, S^n = s^n\} \\
&= \frac{1}{2^{n-k}} \sum_{s^n \in \mathcal{S}} \mathsf{P}\{C^n = v^n \oplus s^n\} \\
&\overset{(a)}{=} \frac{1}{2^n},
\end{aligned}$$

where (a) follows by part (i).

3.B Proof of Lemma 3.1.2

Let $R^n = X^n \oplus V^n$. Thus, $C^n = R^n \oplus \widetilde{H} R^n$. Since R^n is i.i.d. Bern($\frac{1}{2}$), it follows by Lemma 3.1.1 that C^n is uniformly distributed over $\mathcal{C}$. Now, for any $y^n \in \mathcal{Y}^n$ and $u^n \in \{0,1\}^n$, we have

$$\begin{aligned}
\mathsf{P}\{Y^n = y^n, U^n = u^n \,|\, C^n = c^n\} &= \mathsf{P}\{Y^n = y^n, V^n \oplus \widetilde{H} V^n \oplus \widetilde{H} X^n = u^n \,|\, X^n \oplus V^n \oplus \widetilde{H} X^n \oplus \widetilde{H} V^n = c^n\} \\
&= \mathsf{P}\{Y^n = y^n, X^n = c^n \oplus u^n \,|\, X^n \oplus V^n \oplus \widetilde{H}(X^n \oplus V^n) = c^n\} \\
&\overset{(b)}{=} \mathsf{P}\{Y^n = y^n, X^n = c^n \oplus u^n\} \\
&= \prod_{i=1}^n p_{X,Y}(c_i \oplus u_i, y_i) \\
&= \prod_{i=1}^n \bar{p}(y_i, u_i \,|\, c_i),
\end{aligned}$$

where (b) follows since $R^n = X^n \oplus V^n$ is independent of X^n, which implies that C^n and (X^n, Y^n) are independent. This completes the proof.

3.C Proof of Lemma 3.2.1

Denote $\tilde{X}^n = \phi(\tilde{Y}^n, \tilde{V}^n)$. Thus, $\tilde{U}^n = \tilde{X}^n \oplus \tilde{V}^n$. The decoding distance δ of the code (H, ϕ) with respect to the channel $\bar{p}$ can be bounded as follows.

$$\begin{aligned}
\delta &= \frac{1}{2} \sum_{\tilde{y}^n, \tilde{v}^n, \tilde{x}^n} \left| \mathsf{P}\{\tilde{Y}^n = \tilde{y}^n, \tilde{V}^n = \tilde{v}^n, \tilde{X}^n = \tilde{x}^n\} - \frac{1}{2^n} \prod_{i=1}^n \bar{p}(\tilde{y}_i, \tilde{v}_i \,|\, \tilde{x}_i) \right| \\
&= \frac{1}{2} \sum_{\tilde{y}^n, \tilde{v}^n, \tilde{x}^n} \left| \mathsf{P}\{\tilde{Y}^n = \tilde{y}^n, \tilde{V}^n = \tilde{v}^n, \tilde{U}^n = \tilde{x}^n \oplus \tilde{v}^n\} - \frac{1}{2^n} \prod_{i=1}^n p_{X,Y}(\tilde{x}_i \oplus \tilde{v}_i, \tilde{y}_i) \right| \\
&= \frac{1}{2} \sum_{\tilde{y}^n, \tilde{v}^n, \tilde{u}^n} \left| \mathsf{P}\{\tilde{Y}^n = \tilde{y}^n, \tilde{V}^n = \tilde{v}^n, \tilde{U}^n = \tilde{u}^n\} - \frac{1}{2^n} \prod_{i=1}^n p_{X,Y}(\tilde{u}_i, \tilde{x}_i) \right| \\
&\geq \frac{1}{2} \sum_{\tilde{y}^n, \tilde{u}^n} \left| \sum_{\tilde{v}^n} \left(\mathsf{P}\{\tilde{Y}^n = \tilde{y}^n, \tilde{V}^n = \tilde{v}^n, \tilde{U}^n = \tilde{u}^n\} - \frac{1}{2^n} \prod_{i=1}^n p_{X,Y}(\tilde{u}_i, \tilde{y}_i) \right) \right| \\
&= \frac{1}{2} \sum_{\tilde{y}^n, \tilde{u}^n} \left| \mathsf{P}\{\tilde{U}^n = \tilde{u}^n, \tilde{Y}^n = \tilde{y}^n\} - \prod_{i=1}^n p_{X,Y}(\tilde{u}_i, \tilde{y}_i) \right|
\end{aligned}$$

3.D Proof of Lemma 3.2.2

Let $q(x^n, u^n, \hat{x}^n)$ be the joint distribution of $(X^n, U^n, \widehat{X}^n)$, and let $p(x^n, u^n, \hat{x}^n)$ be defined as

$$p(x^n, u^n, \hat{x}^n) = \left(\prod_{i=1}^{n} p_{X,\widehat{X}}(x_i, u_i)\right) \mathbb{1}_{\{\hat{x}^n = u^n\}}.$$

In the block diagram of Figure 3.9, since $U^n = \phi_1(X^n, V^n) \oplus V^n$, we have that $H_1 U^n = H_1 V^n$, and, thus,

$$\begin{bmatrix} \mathbf{0} \\ H_2 U^n \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ H_1 U^n \\ Q U^n \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ H_1 V^n \\ M^{k_1 - k_2} \end{bmatrix},$$

i.e., the input to the lossless source decoder is the index corresponding to U^n. It follows that

$$\begin{aligned}
&\frac{1}{2} \sum_{x^n, \hat{x}^n} |q(x^n, \hat{x}^n) - p(x^n, \hat{x}^n)| \leq \frac{1}{2} \sum_{x^n, u^n, \hat{x}^n} |q(x^n, u^n, \hat{x}^n) - p(x^n, u^n, \hat{x}^n)| \\
&\leq \frac{1}{2} \sum_{x^n, u^n, \hat{x}^n} |q(x^n, u^n) - p(x^n, u^n)|\, q(\hat{x}^n \,|\, x^n, u^n) + \frac{1}{2} \sum_{x^n, u^n, \hat{x}^n} p(x^n, u^n) \left| q(\hat{x}^n \,|\, x^n, u^n) - \mathbb{1}_{\{u^n = \hat{x}^n\}} \right| \\
&= \frac{1}{2} \sum_{x^n, u^n} |q(x^n, u^n) - p(x^n, u^n)| + \frac{1}{2} \sum_{\substack{x^n, u^n, \hat{x}^n: \\ u^n \neq \hat{x}^n}} p(x^n, u^n) q(\hat{x}^n \,|\, x^n, u^n) \\
&\quad + \frac{1}{2} \sum_{x^n, u^n} p(x^n, u^n) \Big(1 - q_{\widehat{X}^n | X^n, U^n}(u^n \,|\, x^n, u^n)\Big) \\
&\overset{(a)}{\leq} \delta + \frac{1}{2}\epsilon + \frac{1}{2}\epsilon \\
&= \delta + \epsilon,
\end{aligned}$$

where (a) follows by Lemma 3.2.1 and the definition of the probability of error of a lossless source code.

Chapter 4

Channel Coding over Networks: A Lego-Brick Approach

In this chapter, we proceed to discuss the channel coding problems for network communication. In Section 4.1, we describe a coding scheme for the Gelfand–Pinsker problem and show that coding for an asymmetric point-to-point channel can follow as a special case of it. In Section 4.2, we construct a Marton coding scheme for broadcast channels and show that it can achieve a corner point of the Marton rate region. In Section 4.3, we describe a coding scheme for multiple access channels. Simulation results for Gelfand–Pinsker coding and Marton coding are provided in Section 4.4.

4.1 Gelfand–Pinsker Coding

In this section, we describe a coding scheme for the binary-input Gelfand–Pinsker problem starting from a Slepian–Wolf code and a point-to-point channel code designed for a BMS channel. Provided that the constituent codes are rate-optimal, the coding scheme can achieve the optimal rate region for the Gelfand–Pinsker problem. First, let's start by describing the Gelfand–Pinsker problem.

4.1.1 Problem Statement

The binary-input Gelfand–Pinsker problem consists of a discrete memoryless channel with state $p(y\,|\,x,s)p(s)$, input alphabet $\mathcal{X} = \{0,1\}$, state alphabet $\mathcal{S}$, output alphabet $\mathcal{Y}$, a collection of conditional probability mass functions $p(y,s\,|\,x)$ on $\mathcal{Y} \times \mathcal{S}$ for each $x \in \mathcal{X}$, and a probability mass function $p(s)$ on $\mathcal{S}$, where the state sequence $(S_1, S_2, \ldots)$ is i.i.d. with $S_i \sim p(s_i)$ and is available noncausally only at the encoder [37]. An (R,n) code (g,ψ) for the Gelfand–Pinsker problem $p(y|x,s)p(s)$ consists of

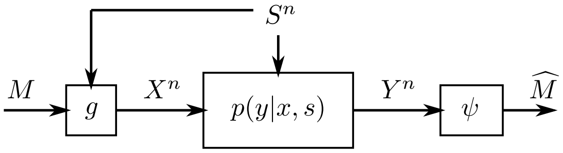


Figure 4.1: A Gelfand–Pinsker code (g, ψ) for a channel with a state that is known noncausally at the encoder.

- a message set $\mathcal{M}$ such that $|\mathcal{M}| = 2^{nR}$,
- an encoder $g : \mathcal{M} \times \mathcal{S}^n \to \mathcal{X}^n$ that assigns a codeword $x^n = g(m, s^n)$ to each message m and state sequence s^n, and
- a decoder $\psi : \mathcal{Y}^n \to \mathcal{M}$ that assigns an estimate $\hat{m} = \psi(y^n)$ to each received sequence y^n.

The average probability of error of the code is $\mathsf{P}\{\widehat{M} \neq M\}$. An (R, n) Gelfand–Pinsker code (g, ψ) is depicted in Figure 4.1.

A rate R is said to be achievable for the Gelfand–Pinsker problem if there exists a sequence of (R, n) Gelfand–Pinsker codes with vanishing error probability asymptotically. The classical result by Gelfand and Pinsker [37] states that any rate

$$R < \max_{p(x|s)} \big(I(X;Y) - I(X;S)\big), \tag{4.1}$$

is achievable for the Gelfand–Pinsker problem. Gelfand and Pinsker's random coding scheme assigns, for each message m and state sequence s^n, a codeword x^n that is jointly typical with s^n for some conditional pmf $p(x|s)$.

4.1.2 Coding Scheme

Consider a binary-input Gelfand–Pinsker problem $p(y\,|\,x, s)p(s)$, and let S^n be an i.i.d. state sequence distributed according to $p(s)$ that is available to the encoder. Inspired by Gelfand and Pinsker's random coding scheme, we construct in the following a coding scheme that shapes the channel input sequence according to some desired conditional distribution $p(x\,|\,s)$of the channel input given the state sequence[1]. At the same time, the channel input sequence should encode the message to the decoder. As we shall see, a key ingredient in achieving these two goals is the nested structure of a pair of linear codes.

More specifically, the Gelfand–Pinsker coding scheme can be constructed from a Slepian–Wolf code and a point-to-point channel code, as described in the following Lego bricks.

[1] For example, $p(x\,|\,s)$ can be chosen to be the maximizer of (4.1).

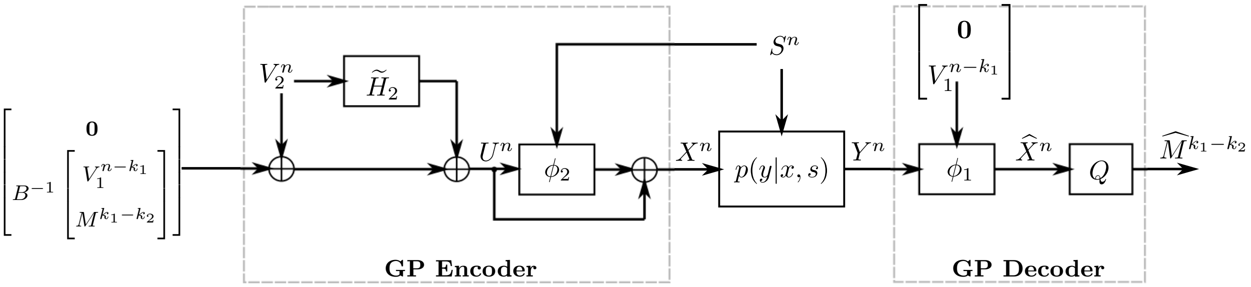


Figure 4.2: A Gelfand–Pinsker coding scheme starting from a point-to-point channel code and a Slepian–Wolf code.

Lego Brick 4.1.1 (SW → GP): *an $(n-k_1, n)$ linear Slepian–Wolf code (H_1, ϕ_1) for the problem*

$$p(x,y) = \sum_s p(s)p(x\,|\,s)p(y\,|\,x,s).$$

Let $\mathcal{C}_1$ be the codebook corresponding to H_1, and let ϵ be the average probability of error of the Slepian–Wolf code.

Lego Brick 4.1.2 (P2P → GP): *a (k_2, n) linear point-to-point channel code (H_2, ϕ_2) with codebook $\mathcal{C}_2$ for the channel*

$$\bar{p}(s, v\,|\,x) = p_{X,S}(x \oplus v, s). \tag{4.2}$$

Let δ denote the decoding distance of the code (H_2, ϕ_2) with respect to the channel $\bar{p}$. Furthermore, we assume that the two codes are nested, i.e., $\mathcal{C}_2 \subseteq \mathcal{C}_1$.

Remark 4.1.1. Since $\mathcal{C}_2 \subseteq \mathcal{C}_1$, we will assume, without loss of generality, that H_1 is a submatrix of H_2, i.e., $H_2 = \begin{bmatrix} H_1 \\ Q \end{bmatrix}$ for some $(k_1 - k_2) \times n$ matrix Q. Further, let $H_2 = \begin{bmatrix} A & B \end{bmatrix}$, where B is nonsingular, and let $\widetilde{H}_2$ be as defined in Remark 2.1.1.

Remark 4.1.2. The channel $\bar{p}$ in (4.2) is the symmetrized channel corresponding to the desired joint distribution $p(x,s) = p(s)p(x\,|\,s)$.

Figure 4.2 shows the block diagram of the Gelfand–Pinsker coding scheme, where $V_1^{n-k_1}$ is an i.i.d. Bern(1/2) random dither shared between the encoder and the decoder, and V_2^n is an i.i.d. Bern(1/2) sequence generated independently at the encoder (not necessarily shared with the decoder). To simplify the notation, let us denote the input to the Gelfand–Pinsker encoder in Figure 4.2 by

$$Z^n \triangleq \begin{bmatrix} \mathbf{0} \\ B^{-1} \begin{bmatrix} V_1^{n-k_1} \\ M^{k_1-k_2} \end{bmatrix} \end{bmatrix}. \tag{4.3}$$

The coding scheme uses the decoding function ϕ_2 to shape the channel input X^n according to the desired distribution. To see why this holds in the construction of Figure 4.2, let $U^n = Z^n \oplus V_2^n \oplus \widetilde{H}_2 V_2^n$ denote one of the two inputs to ϕ_2, where the second input is S^n. Notice that by Lemma 3.1.1, $V_2^n \oplus \widetilde{H}_2 V_2^n \sim \mathrm{Unif}(\mathcal{C}_2)$, and U^n is i.i.d. Bern(1/2) (and independent of S^n). Hence, by Lemma 3.2.1, the sequence $X^n = \phi_2(S^n, U^n) \oplus U^n$ satisfies that

$$\frac{1}{2} \sum_{x^n, s^n} \left| \mathsf{P}\{X^n = x^n, S^n = s^n\} - \prod_{i=1}^{n} p(x_i, s_i) \right| \leq \delta, \tag{4.4}$$

and thus, the distribution of (X^n, S^n) is δ-away in total variation distance from the desired joint distribution.

Moreover, it holds that

$$\begin{bmatrix} H_1 X^n \\ Q X^n \end{bmatrix} = H_2 X^n = H_2 U^n \overset{(a)}{=} H_2 Z^n = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ B^{-1} \begin{bmatrix} V_1^{n-k_1} \\ M^{k_1-k_2} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} V_1^{n-k_1} \\ M^{k_1-k_2} \end{bmatrix},$$

where (a) follows since $V_2^n \oplus \widetilde{H}_2 V_2^n \in \mathcal{C}_2$. Therefore, $H_1 X^n = V_1^{n-k_1}$, which is the index inputted to the Slepian–Wolf decoder ϕ_1 at the decoder side.

Remark 4.1.3. Intuitively, the Gelfand–Pinsker construction can be understood as follows. The sequence $V_1^{n-k_1}$ represents a coset shift of the outer code $\mathcal{C}_1$, whereas the message $M^{k_1-k_2}$ represents a coset shift of the inner code $\mathcal{C}_2$ within the outer code. Since $H_2 X^n = H_2 Z^n$ (by construction), the channel input X^n belongs to the coset of the inner code $\mathcal{C}_2$ indexed by $(V_1^{n-k_1}, M^{k_1-k_2})$. Since the sequence $V_1^{n-k_1}$ is shared between the encoder and the decoder, the coset shift with respect to the outer code $\mathcal{C}_1$ is known to the decoder, which can be leveraged by the Slepian–Wolf decoder to recover an estimate of the channel input sequence, and, hence, the message. Note that the idea of using such a nested structure to code over a Gelfand–Pinsker channel has been considered in [83], where a joint typicality encoder and decoder was used.

Therefore, the coding scheme can be summarized as follows.

Encoding: To transmit the message $m^{k_1-k_2}$ upon observing the state sequence s^n, the encoder computes the sequence z^n as in (4.3) using a random dither $v_1^{n-k_1}$ shared with the decoder, and sends $x^n = \phi_2(s^n, z^n \oplus v_2^n \oplus \widetilde{H}_2 v_2^n) \oplus z^n \oplus v_2^n \oplus \widetilde{H}_2 v_2^n$ over the channel, where v_2^n is a random dither generated independently at the encoder.

Decoding: Upon observing the channel output y^n, the decoder computes $\hat{x}^n = \phi_1 \left(\begin{bmatrix} \mathbf{0} \\ v_1^{n-k_1} \end{bmatrix}, y^n \right)$, and declares $\hat{m}^{k_1-k_2} = Q\hat{x}^n$ as the message estimate.

Analysis of the probability of error: Let $q(x^n, s^n)$ be the distribution of (X^n, S^n) in Figure 4.2, and let $p(x^n, s^n)$ be the i.i.d. distribution according to $p(x, s)$. The average probability of error can be bounded as

$$\begin{aligned}
\mathsf{P}\{\widehat{M}^{k_1-k_2} \neq M^{k_1-k_2}\} &\leq \mathsf{P}\{\widehat{X}^n \neq X^n\} \\
&= \sum_{x^n,s^n} \mathsf{P}\{\widehat{X}^n \neq X^n | X^n = x^n, S^n = s^n\} q(x^n, s^n) \\
&= \sum_{x^n,s^n} \mathsf{P}\{\widehat{X}^n \neq X^n | X^n = x^n, S^n = s^n\} \big(q(x^n, s^n) - p(x^n, s^n)\big) \\
&\quad + \sum_{x^n,s^n} \mathsf{P}\{\widehat{X}^n \neq X^n | X^n = x^n, S^n = s^n\} p(x^n, s^n) \\
&\overset{(a)}{\leq} \frac{1}{2} \sum_{x^n,s^n} \big|q(x^n, s^n) - p(x^n, s^n)\big| + \epsilon \\
&\overset{(b)}{\leq} \delta + \epsilon,
\end{aligned}$$

where (a) holds since $\sum_i c_i(a_i - b_i) \leq \frac{1}{2}\sum_i |a_i - b_i|$ whenever $0 \leq c_i \leq 1$ and $\sum_i a_i = \sum_i b_i$ and by the fact that $\mathsf{P}\{\widehat{X}^n \neq X^n | X^n = x^n, S^n = s^n\}$ depends only on the channel $p(y|x, s)$, and (b) follows by equation (4.4).

Rate: The rate of the coding scheme is $R = \frac{k_1 - k_2}{n}$.

Remark 4.1.4. Recall from Section 2.3 and Remark 2.2.1 that a sequence of point-to-point channel codes with a vanishing decoding distance δ over the channel $\bar{p}$ defined in (4.2) exists if (and only if) the rate is larger than $1 - H(X\,|\,S)$.

Remark 4.1.5. If the rate of the point-to-point channel code is $\frac{k_2}{n} = 1 - H(X\,|\,S) + \gamma_1$ for some $\gamma_1 > 0$, and the rate of the Slepian–Wolf code is $\frac{n-k_1}{n} = H(X\,|\,Y) + \gamma_2$ for some $\gamma_2 > 0$, then the rate of the Gelfand–Pinsker coding scheme is

$$R = \frac{k_1 - k_2}{n} = H(X\,|\,S) - H(X\,|\,Y) - \gamma_1 - \gamma_2 = I(X;Y) - I(X;S) - \gamma_1 - \gamma_2.$$

Conclusion: Starting from an $(n - k_1, n)$ Slepian–Wolf code with an average probability of error ϵ and a (k_2, n) linear point-to-point channel code with decoding distance δ, we have constructed a $(\frac{k_1-k_2}{n}, n)$ Gelfand–Pinsker code that targets a conditional distribution $p(x\,|\,s)$ of the channel input given the channel state and has an average probability of error that is bounded by $\delta + \epsilon$.

4.1.3 Specialization to Asymmetric Channel Coding

The problem of coding for an asymmetric point-to-point channel (Section 2.1) can be seen as a special case of the Gelfand–Pinkser coding problem when the channel state is constant and independent of the channel output. This observation has been previously formalized in [55]. For completion, we give the details here, along with the resulting construction of an asymmetric channel coding scheme.

Suppose we have an (R, n) code (g, ψ) for the binary-input Gelfand–Pinsker problem $p(y \,|\, x, s)p(s)$, where $p(y \,|\, x, s) = p(y \,|\, x)$ (i.e., the channel output is independent of the state given the channel input) and $p_S(0) = 1$. Let ϵ be the average probability of error of the code. Define $f : [2^{nR}] \to \{0,1\}^n$ by $f(m) = g(m, \mathbf{0})$. Then, (f, ψ) forms a code for the channel $p(y \,|\, x)$ with length n, rate R and average probability of error

$$\begin{aligned}
&\sum_m 2^{-nR}. \mathsf{P}\{\psi(Y^n) \neq m \,|\, X^n = f(m)\} = \sum_m 2^{-nR}. \mathsf{P}\{\psi(Y^n) \neq m \,|\, X^n {=} g(m, \mathbf{0})\} \\
&= \sum_m 2^{-nR} \Big(\mathsf{P}\{\psi(Y^n) \neq m \,|\, X^n {=} g(m, \mathbf{0}), S^n {=} \mathbf{0}\} \prod_{i=1}^n p_S(0) \\
&\qquad\qquad + \sum_{s^n \neq \mathbf{0}} \mathsf{P}\{\psi(Y^n) \neq m \,|\, X^n {=} g(m, \mathbf{0}), S^n {=} s^n\} \prod_{i=1}^n p_S(s_i) \Big) \\
&\overset{(a)}{=} \sum_m 2^{-nR}\, \mathsf{P}\{\psi(Y^n) \neq m \,|\, X^n {=} g(m, \mathbf{0}), S^n {=} \mathbf{0}\} \\
&\overset{(b)}{=} \sum_m 2^{-nR} \Big(\mathsf{P}\{\psi(Y^n) \neq m \,|\, X^n {=} g(m, \mathbf{0}), S^n {=} \mathbf{0}\} \prod_{i=1}^n p_S(0) \\
&\qquad\qquad + \sum_{s^n \neq \mathbf{0}} \mathsf{P}\{\psi(Y^n) \neq m \,|\, X^n {=} g(m, s^n), S^n {=} s^n\} \prod_{i=1}^n p_S(s_i) \Big) \\
&= \sum_{m, s^n} 2^{-nR} \Big(\prod_{i=1}^n p_S(s_i) \Big) \mathsf{P}\{\psi(Y^n) \neq m \,|\, X^n {=} g(m, s^n), S^n {=} s^n\} \\
&= \epsilon,
\end{aligned}$$

where (a) and (b) follow since $\prod_{i=1}^n p_S(s_i) = 0$ for $s^n \neq \mathbf{0}$.

Therefore, the Gelfand–Pinsker coding scheme described in Section 4.1.2 can be used to construct an asymmetric channel code in the special case when the state sequence is the constant all-zero sequence, i.e., $p_S(0) = 1$. To this end, consider a binary-input channel $p(y \,|\, x)$ that is not necessarily symmetric. Let the capacity-achieving input distribution be $p(x) \sim \mathrm{Bern}(\alpha)$ for some $\alpha \in (0, 1/2)$ (i.e., a non-uniform distribution). By specializing the Lego bricks of the Gelfand–Pinsker coding scheme to the case when $p_S(0) = 1$ and the desired $p(x \,|\, s)$ is $\mathrm{BSC}(\alpha)$, the asymmetric channel coding scheme can be constructed using the following Lego bricks.

Lego Brick 4.1.3 (SW → Asym): *an $(n - k_1, n)$ linear Slepian–Wolf code (H_1, ϕ_1) for the problem $p(x, y) = p(x)p(y \,|\, x)$ with codebook $\mathcal{C}_1$ and average probability of error ϵ.*

Lego Brick 4.1.4 (P2P → Asym): *a (k_2, n) linear point-to-point channel code (H_2, ϕ_2) for* $\mathrm{BSC}(\alpha)$ *with codebook $\mathcal{C}_2$ and decoding distance δ. Furthermore, we assume that the two codes are nested, i.e., $\mathcal{C}_2 \subseteq \mathcal{C}_1$.*

Remark 4.1.6. As before, it is assumed that H_1 is a submatrix of H_2, i.e., $H_2 = \begin{bmatrix} H_1 \\ Q \end{bmatrix}$ for some $(k_1 - k_2) \times n$ matrix Q, and $H_2 = \begin{bmatrix} A & B \end{bmatrix}$, for some non-singular matrix B.

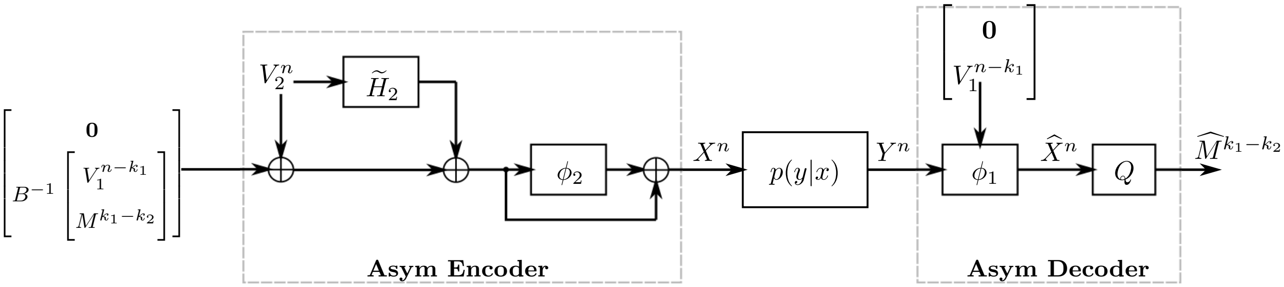


Figure 4.3: A coding scheme for an asymmetric channel starting from a Slepian–Wolf code and a point-to-point channel code.

Figure 4.3 shows the block diagram for the asymmetric channel coding scheme, which can be seen as a specialization of the Gelfand–Pinsker coding scheme when the state sequence is the constant all-zero sequence. The description of the coding scheme and the analysis of its probability of error follow similarly as in the Gelfand–Pinsker case.

Remark 4.1.7. If the rate of the point-to-point channel code is $\frac{k_2}{n} = 1 - H(X) + \gamma_1$ for some $\gamma_1 > 0$, and the rate of the Slepian–Wolf code is $\frac{n-k_1}{n} = H(X\,|\,Y) + \gamma_2$ for some $\gamma_2 > 0$, then the rate of the asymmetric channel coding scheme is

$$\frac{k_1 - k_2}{n} = I(X;Y) - \gamma_1 - \gamma_2.$$

Remark 4.1.8. The asymmetric channel encoder shown in Figure 4.3 is almost identical to the lossy source decoder shown in Figure 3.9 (when the constituent lossless source decoder is implemented using a point-to-point channel code). Similarly, the lossy source encoder is almost identical to the asymmetric channel decoder (when the constituent Slepian–Wolf decoder is implemented using a point-to-point channel code). This suggests some form of duality between the two constructions. A similar observation can be made between the Gelfand–Pinsker and Wyner–Ziv constructions.

Conclusion: Starting from an $(n - k_1, n)$ Slepian–Wolf code with an average probability of error ϵ and a (k_2, n) linear point-to-point channel code with decoding distance δ, we have constructed a $(k_1 - k_2, n)$ code for an asymmetric point-to-point channel that targets an input distribution $p(x)$ and has an average probability of error that is bounded by $\delta + \epsilon$.

4.2 Marton Coding over Broadcast Channels

In this section, we construct a Marton coding scheme for the K-user broadcast channel with K transmit antennas starting from an asymmetric channel code and a Gelfand–Pinsker code. The coding scheme that we present can achieve a corner point in the Marton coding achievable rate region. First, let us review the channel coding problem over a broadcast channel.

4.2.1 Problem Statement

A binary-input discrete memoryless broadcast channel with K users and K transmit antennas consists of a binary input alphabet $\mathcal{X} = \{0,1\}$, output alphabets $\mathcal{Y}_1, \ldots, \mathcal{Y}_K$, and a collection of conditional probability mass functions $p(y_1^K \,|\, x_1^K)$ on $\mathcal{Y}_1 \times \cdots \times \mathcal{Y}_K$ for each $(x_1, \ldots, x_K) \in \mathcal{X}^K$. An $(R_1, \ldots, R_K, n)$ code $(g, \psi_1, \ldots, \psi_K)$ for the channel $p(y_1^K \,|\, x_1^K)$ consists of

- message sets $\mathcal{M}_1, \ldots, \mathcal{M}_K$ such that $|\mathcal{M}_j| = 2^{nR_j}$ for each $j \in [K]$,
- an encoder $g : \mathcal{M}_1 \times \cdots \times \mathcal{M}_K \to \mathcal{X}^{Kn}$ that maps each message tuple $(m_1, \ldots, m_K)$ to K codewords $(x_1^n, \ldots, x_K^n) = g(m_1, \ldots, m_K)$,
- decoders $\psi_j : \mathcal{Y}_j^n \to \mathcal{M}_j$, for $j \in [K]$, that assign message estimates $\hat{m}_j = \psi_j(y_j^n)$ to each received sequence y_j^n.

We say that the rate of the code is the tuple $(R_1, \ldots, R_K)$, and its *sum-rate* is $R_{\text{sum}} = R_1 + \cdots + R_K$. The average probability of error of the code over the channel is defined as

$$\epsilon = \mathsf{P}\left\{\widehat{M}_j \neq M_j \text{ for some } j \in [K]\right\}.$$

A rate tuple $(R_1, \ldots, R_K)$ is said to be achievable for the broadcast channel if there exists a sequence of $(R_1, \ldots, R_K, n)$ codes with vanishing error probability asymptotically. Marton [41] described a random coding scheme for the broadcast channel that assigns, for each message tuple $(m_1, \ldots, m_K)$, a sequence $(x_1^n, \ldots, x_K^n)$ that are jointly typical for some input distribution $p(x_1, \ldots, x_K)$. For such a distribution, the achievable rate region of Marton's coding scheme is the set of rate tuples $(R_1, \ldots, R_K)$ such that

$$R(\mathcal{S}) < \sum_{j \in \mathcal{S}} I(X_j; Y_j) - I^*(X_{\mathcal{S}}),$$

for all $\mathcal{S} \subseteq [K]$, where $R(\mathcal{S}) = \sum_{j \in \mathcal{S}} R_j$, $X_{\mathcal{S}} = (X_j : j \in \mathcal{S})$, and $I^*(X_{\mathcal{S}}) = \sum_{j \in \mathcal{S}} I(X_j; X_{[j-1] \cap \mathcal{S}})$. For example, for $K = 2$ and an input distribution $p(x_1, x_2)$, the achievable rate region is the set of rate pairs (R_1, R_2) such that

$$\begin{aligned} R_1 &< I(X_1; Y_1), \\ R_2 &< I(X_2; Y_2), \\ R_1 + R_2 &< I(X_1; Y_1) + I(X_2; Y_2) - I(X_1; X_2). \end{aligned} \tag{4.5}$$

4.2.2 Coding Scheme

Consider a two-user broadcast channel $p(y_1, y_@ \,|\, x_1, x_2)$ with two transmit antennas. Inspired by Marton's coding scheme, the proposed construction targets an input distribution $p(x_1, x_2)$ and uses the following asymmetric channel code and Gelfand–Pinsker code as its constituent Lego bricks.

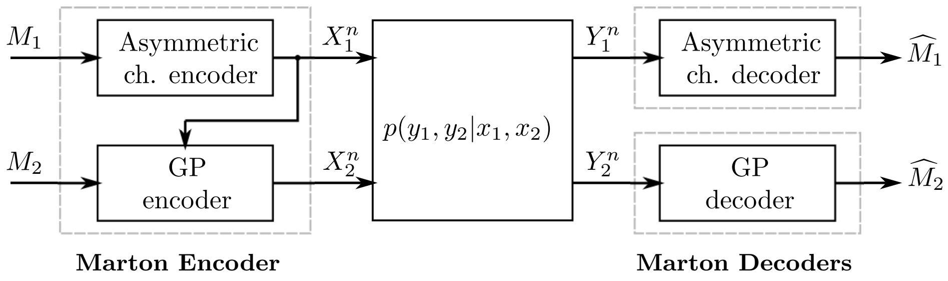


Figure 4.4: Marton coding for the two-user broadcast channel using an asymmetric channel code and a Gelfand–Pinsker code.

Lego Brick 4.2.1 (Asym → Marton): *an (R_1, n) asymmetric channel code (g_1, ψ_1) for the channel*

$$p(y_1 \,|\, x_1) = \sum_{y_2, x_2} p(x_2 \,|\, x_1) p(y_1, y_2 \,|\, x_1, x_2),$$

which targets an input distribution $p(x_1)$, such that, for $M_1 \sim \mathrm{Unif}([2^{nR_1}])$, the channel input $X_1^n = g_1(M_1)$ satisfies

$$\frac{1}{2} \sum_{x_1^n} \left| P\{X_1^n = x_1^n\} - \prod_{i=1}^n p(x_{1i}) \right| \leq \delta_1, \tag{4.6}$$

for some $\delta_1 > 0$. Let ϵ_1 be the average probability of error of the asymmetric channel code when the channel input distribution is i.i.d. according to $p(x_1)$.

Lego Brick 4.2.2 (GP → Marton): *an (R_2, n) code (g_2, ψ_2) for the Gelfand–Pinsker problem defined by*

$$p(x_1) p(y_2 \,|\, x_2, x_1) = p(x_1) \sum_{y_1} p(y_1, y_2 \,|\, x_1, x_2),$$

which targets a conditional distribution $p(x_2 \,|\, x_1)$, such that, when $M_2 \sim \mathrm{Unif}([2^{nR_2}])$ and $\widetilde{X}_1^n$ is i.i.d. $p(x_1)$ sequence, the channel input $\widetilde{X}_2^n = g_2(M_2, \widetilde{X}_1^n)$ satisfies

$$\frac{1}{2} \sum_{x_1^n, x_2^n} \left| P\{\widetilde{X}_1^n = x_1^n, \widetilde{X}_2^n = x_2^n\} - \prod_{i=1}^n p(x_{1i}, x_{2i}) \right| \leq \delta_2, \tag{4.7}$$

for some $\delta_2 > 0$. Let ϵ_2 be the average probability of error of the Gelfand–Pinsker code when the conditional distribution of the channel input given the channel state is i.i.d. according to $p(x_2 \,|\, x_1)$.

Remark 4.2.1. Each of the asymmetric channel code and the Gelfand–Pinsker code can be constructed starting from a point-to-point channel code and a Slepian–Wolf code, as described in Section 4.1.

Figure 4.4 shows the block diagram of the Marton coding scheme for the broadcast channel. The main idea is to shape the channel input according to the desired distribution. From conditions (4.6) and (4.7), it follows from standard arguments that

$$\frac{1}{2}\sum_{x_1^n, x_2^n}\left|\mathsf{P}\{X_1^n = x_1^n, X_2^n = x_2^n\} - \prod_{i=1}^{n} p(x_{1i}, x_{2i})\right| \leq \delta_1 + \delta_2,$$

which says that the channel input distribution is $(\delta_1 + \delta_2)$-away in total variation distance from the desired target distribution. The average probability of error of the coding scheme can be bounded as

$$\mathsf{P}\left\{\{\widehat{M}_1 \neq M_1\} \cup \{\widehat{M}_2 \neq M_2\}\right\} \leq \delta_1 + \delta_2 + \epsilon_1 + \epsilon_2,$$

which follows by a similar analysis as in Section 4.1. The rate of the coding scheme is the pair (R_1, R_2).

Remark 4.2.2. If the rate of the asymmetric channel code is $R_1 = I(X_1; Y_1) - \gamma_1$ for some $\gamma_1 > 0$, and the rate of the Gelfand–Pinsker code is $R_2 = I(X_2; Y_2) - I(X_1; X_2) - \gamma_2$ for some $\gamma_2 > 0$, it follows that the rate of the Marton coding scheme is

$$(R_1, R_2) = (I(X_1; Y_1) - \gamma_1,\, I(X_2; Y_2) - I(X_1; X_2) - \gamma_2)\,.$$

Note that the rate pair $(I(X_1; Y_1),\, I(X_2; Y_2) - I(X_1; X_2))$ is a corner point of Marton's rate region for the broadcast channel, given in (4.5). If the encoding order is reversed (i.e., X_2^n is encoded using an asymmetric channel code and used as a state sequence to encode M_1), another corner point of the rate region can be achieved.

Remark 4.2.3. A similar coding scheme can be constructed for a K-user broadcast channel using one asymmetric channel code and $K - 1$ Gelfand–Pinsker codes, where the encoder successively encodes the input sequences.

Conclusion: Starting from an asymmetric channel code and a Gelfand–Pinsker code, we have constructed a Marton code for the K-user broadcast.

4.3 Coding for Multiple Access Channels

In this section, we describe a coding scheme for the multiple access channel that uses two asymmetric channel codes and successive cancellation decoding. The coding scheme can achieve the corner points of the optimal rate region for the multiple access channel.

4.3.1 Problem Statement

We consider the problem of coding for a binary-input discrete memoryless multiple access channel consisting of input alphabets $\mathcal{X}_1 = \mathcal{X}_2 = \{0, 1\}$, an arbitrary output alphabet $\mathcal{Y}$ and a collection of conditional probability mass functions $p(y \,|\, x_1, x_2)$ for each $(x_1, x_2) \in \mathcal{X}_1 \times \mathcal{X}_2$. An (R_1, R_2, n) code for the multiple access channel consists of

- two message sets $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $|\mathcal{M}_j| = 2^{nR_j}$, $j = 1, 2$,
- encoders $g_j : \mathcal{M}_j \to \mathcal{X}_j^n$ for $j = 1, 2$, that map each message m_j to a codeword x_j^n,
- a decoder $\psi : \mathcal{Y}^n \to \mathcal{M}_1 \times \mathcal{M}_2$ that assigns message estimates $(\hat{m}_1, \hat{m}_2)$ to each received sequence y^n.

The average probability of error of the code is $\epsilon = \mathsf{P}\left\{\{\widehat{M_1} \neq M_1\} \cup \{\widehat{M_2} \neq M_2\}\right\}$. A rate pair (R_1, R_2) is said to be achievable for the multiple access channel if there exists a sequence of (R_1, R_2, n) codes with vanishing error probability asymptotically. It is well-known that a rate pair (R_1, R_2) is achievable if

$$\begin{aligned} R_1 &< I(X_1; Y \,|\, X_2), \\ R_2 &< I(X_2; Y \,|\, X_1), \\ R_1 + R_2 &< I(X_1, X_2; Y), \end{aligned} \tag{4.8}$$

for some pmf $p(x_1)p(x_2)$. The multiple access channel coding problem was first alluded to by Shannon [40], and the characterization of the optimal rate region is due to Ahlswede [84], and Slepian and Wolf [85]. In the following, we will show how two asymmetric channel codes can be used to achieve a corner point of the rate region (4.8).

4.3.2 Coding Scheme

The coding scheme for the multiple access channel uses two asymmetric channel codes that target two desired input pmf's $p(x_1)$ and $p(x_2)$, and employs a successive cancellation decoder to decode the messages at the receiver side. Hence, the coding scheme for the multiple access channel can be constructed using the following two Lego bricks.

Lego Brick 4.3.1 (Asym → MAC): *an (R_1, n) asymmetric channel code (g_1, ψ_1) for the channel*

$$p(y \,|\, x_1) = \sum_{x_2} p(x_2) p(y \,|\, x_1, x_2),$$

which targets an input distribution $p(x_1)$, such that, for $M_1 \sim \text{Unif}([2^{nR_1}])$, the channel input $X_1^n = g_1(M_1)$ satisfies

$$\frac{1}{2} \sum_{x_1^n} \left| P\{X_1^n = x_1^n\} - \prod_{i=1}^{n} p(x_{1i}) \right| \leq \delta_1, \tag{4.9}$$

for some $\delta_1 > 0$. Let ϵ_1 be the average probability of error of the asymmetric channel code when the channel input distribution is i.i.d. according to $p(x_1)$.

Lego Brick 4.3.2 (Asym → MAC): *an (R_2, n) asymmetric channel code (g_2, ψ_2) for the channel*

$$p(y, x_1 \,|\, x_2) = p(x_1) p(y \,|\, x_1, x_2),$$

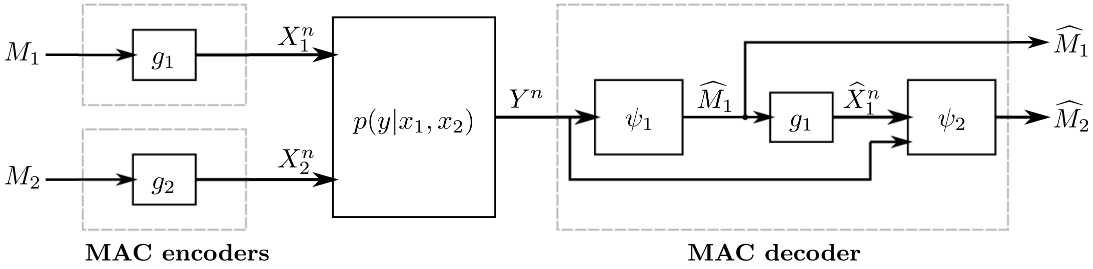


Figure 4.5: Coding scheme for multiple access channel using two asymmetric channel codes.

which targets an input distribution $p(x_2)$, such that, for $M_2 \sim \mathrm{Unif}([2^{nR_2}])$, the channel input $X_2^n = g_2(M_2)$ satisfies

$$\frac{1}{2}\sum_{x_2^n}\left|\mathsf{P}\{X_2^n = x_2^n\} - \prod_{i=1}^{n} p(x_{2i})\right| \leq \delta_2, \tag{4.10}$$

for some $\delta_2 > 0$. Let ϵ_2 be the average probability of error of the asymmetric channel code when the channel input distribution is i.i.d. according to $p(x_2)$.

Remark 4.3.1. Each of the asymmetric channel codes can be constructed starting from a point-to-point channel code and a Slepian–Wolf code, as described in Section 4.1.3.

Figure 4.5 shows the block diagram of the coding scheme. Inspired by the random coding scheme for the multiple access channel [85], the two asymmetric channel encoders are used to target the desired joint pmf's $p(x_1)$ and $p(x_2)$ at the encoder side, and the decoder recovers estimates of the messages sequentially, as per successive cancellation decoding. At the encoder side, conditions (4.9) and (4.10) on the asymmetric channel codes guarantee that

$$\frac{1}{2}\sum_{x_j^n}\left|\mathsf{P}\{X_j^n = x_j^n\} - \prod_{i=1}^{n} p(x_{ji})\right| \leq \delta_j,$$

for each $j = 1, 2$. The coding scheme can be summarized as follows.

Encoding: The two encoders encode their messages using the point-to-point channel encoders, i.e., encoder 1 transmits $x_1^n = g_1(m_1)$ upon observing the message m_1, and encoder 2 transmits $x_2^n = g_2(m_2)$ upon observing the message m_2.

Decoding: Upon observing the output sequence y^n, the decoder declares $\hat{m}_1 = \psi_1(y^n)$ as the estimate of the first user's message. Then, the decoder computes $\hat{x}_1^n = g_1(\hat{m}_1)$ and declares $\hat{m}_2 = \psi_2(y^n, \hat{x}_1^n)$ as the estimate of the second user's message.

Analysis of probability of error: Denote by $q(x_j^n)$ the distribution of X_j^n and by $p(x_j^n)$ the i.i.d. distribution according to $p(x_j)$, for $j = 1, 2$. Define $\widehat{X}_j^n = g_j(\widehat{M}_j)$, for $j = 1, 2$. Then the average probability of error of the coding scheme can be bounded as

$$\begin{aligned}
&\mathsf{P}\left\{\{\widehat{M}_1 \neq M_1\} \cup \{\widehat{M}_2 \neq M_2\}\right\} = \mathsf{P}\left\{\{\widehat{X}_1^n \neq X_1^n\} \cup \{\widehat{X}_2^n \neq X_2^n\}\right\} \\
&= \sum_{x_1^n, x_2^n} q(x_1^n) q(x_2^n)\, \mathsf{P}\left\{\{\widehat{X}_1^n \neq X_1^n\} \cup \{\widehat{X}_2^n \neq X_2^n\} \,|\, X_1^n = x_1^n, X_2^n = x_2^n\right\} \\
&\overset{(a)}{\leq} \sum_{x_1^n, x_2^n} p(x_1^n) p(x_2^n)\, \mathsf{P}\left\{\{\widehat{X}_1^n \neq X_1^n\} \cup \{\widehat{X}_2^n \neq X_2^n\} \,|\, X_1^n = x_1^n, X_2^n = x_2^n\right\} \\
&\qquad + \frac{1}{2} \sum_{x_1^n, x_2^n} |q(x_1^n) q(x_2^n) - p(x_1^n) p(x_2^n)| \\
&\overset{(b)}{\leq} \sum_{x_1^n, x_2^n} p(x_1^n) p(x_2^n)\, \mathsf{P}\{\widehat{X}_1^n \neq X_1^n \,|\, X_1^n = x_1^n, X_2^n = x_2^n\} \\
&\qquad + \sum_{x_1^n, x_2^n} p(x_1^n) p(x_2^n)\, \mathsf{P}\{\widehat{X}_2^n \neq X_2^n \,|\, X_1^n = x_1^n, X_2^n = x_2^n, \widehat{X}_1^n = x_1^n\} \\
&\qquad + \frac{1}{2} \sum_{x_1^n} |q(x_1^n) - p(x_1^n)| + \frac{1}{2} \sum_{x_2^n} |q(x_2^n) - p(x_2^n)| \\
&\overset{(c)}{\leq} \epsilon_1 + \epsilon_2 + \delta_1 + \delta_2,
\end{aligned}$$

where (a) holds since $\sum_i c_i(a_i - b_i) \leq \frac{1}{2} \sum_i |a_i - b_i|$ whenever $0 \leq c_i \leq 1$ and $\sum_i a_i = \sum_i b_i$, (b) holds by the union bound, and (c) holds by conditions (4.9) and (4.10) and the definition of average probability of error.

Rate: The coding scheme attains the rate pair (R_1, R_2).

Remark 4.3.2. If the rate of the first asymmetric channel code is $R_1 = I(X_1; Y) - \gamma_1$ for some $\gamma_1 > 0$, and the rate of the second asymmetric channel code is $R_2 = I(X_2; X_1, Y) - \gamma_2 = I(X_2; Y \,|\, X_1) - \gamma_2$ for some $\gamma_2 > 0$, then the coding scheme attains the rate pair

$$(R_1, R_2) = (I(X_1; Y) - \gamma_1,\ I(X_2; Y \,|\, X_1) - \gamma_2)\,.$$

Note that the rate pair $(I(X_1; Y),\ I(X_2; Y \,|\, X_1))$ is a corner point of the rate region given in (4.8). By reversing the decoding order and appropriately modifying the channels that the two asymmetric channel codes are designed for, another corner point of the rate region can be achieved.

Remark 4.3.3. A similar coding scheme can be constructed for the K-user multiple access channel using K asymmetric channel codes, where the decoder successively decodes the user messages.

Conclusion: Starting from K asymmetric channel codes, we have constructed a code for the K-user multiple access channel.

4.4 Simulation Results

4.4.1 Gelfand–Pinsker Coding

The Gelfand–Pinsker coding scheme described in Section 4.1 can be simulated using two point-to-point channel codes. This can be done by constructing the Slepian–Wolf code (H_1, ϕ_1) using its constituent point-to-point code as described in Section 3.1.2. If R_1 and R_2 are the rates of the two point-to-point codes, it follows that the rate of the Gelfand–Pinsker coding scheme is $R = R_1 - R_2$.

We consider a binary-input Gaussian channel with a binary state whose channel output can be expressed as

$$Y = X + gS + Z,$$

where $X \in \{-\sqrt{P}, \sqrt{P}\}$ is the channel input, $S \in \{-\sqrt{P}, \sqrt{P}\}$ is a channel state that is known noncausally only at the encoder with $\mathsf{P}\{S = -\sqrt{P}\} = \theta$, and $Z \sim \mathcal{N}(0,1)$ is a sample from an i.i.d. Gaussian noise process. In our simulations, we use $g = 0.9$ and $\theta = 0.1$.

The proposed Gelfand–Pinsker coding scheme can be used to code over the given channel model. We use polar codes with successive cancellation decoding as the constituent point-to-point channel codes. To construct the polar codes (i.e., identify the information sets), we use Arıkan's method of sorting upper bounds on the Bhattacharyya parameters of the synthetic polar bit-channels [6]. To ensure nestedness, the information set of the polar code corresponding to (H_2, ϕ_2) is taken to be a subset of that corresponding to the code (H_1, ϕ_1). Inspired by the characterization (4.1) of achievable rates for the Gelfand–Pinsker problem, the proposed coding scheme targets the the conditional distribution $p^*(x \,|\, s)$ that maximizes $I(X;Y) - I(X;S)$, i.e.,

$$p^*(x \,|\, s) = \underset{p(x|s)}{\arg\max} \left(I(X;Y) - I(X;S) \right). \tag{4.11}$$

Since X and S are binary, the optimization (4.11) can be solved efficiently using off-the-shelf numerical solvers (or simply using a grid search over the two parameters that define $p(x \,|\, s)$). The rates R_1 and R_2 are chosen "close" to their theoretical limits (see Remark 4.1.5) when the conditional distribution of the channel input given the channel state is $p^*(x \,|\, s)$. More precisely, we take

$$\begin{aligned} R_1 &= 1 - H(X \,|\, Y) - \gamma, \\ R_2 &= 1 - H(X \,|\, S), \end{aligned} \tag{4.12}$$

where $\gamma > 0$ is a "back-off" parameter from the theoretical limit, which allows for a reasonable error probability for the code (H_1, ϕ_1). In our Gelfand–Pinsker simulations, we used $\gamma = 3/16$.

We compare the proposed coding scheme with the simple strategy that encodes the transmitter's message using a point-to-point channel code while ignoring the available state sequence. Since the induced channel input distribution would be Bern(1/2) in this case, we refer to this strategy as the "symmetric coding" strategy. For a fair comparison, we also use a polar code with

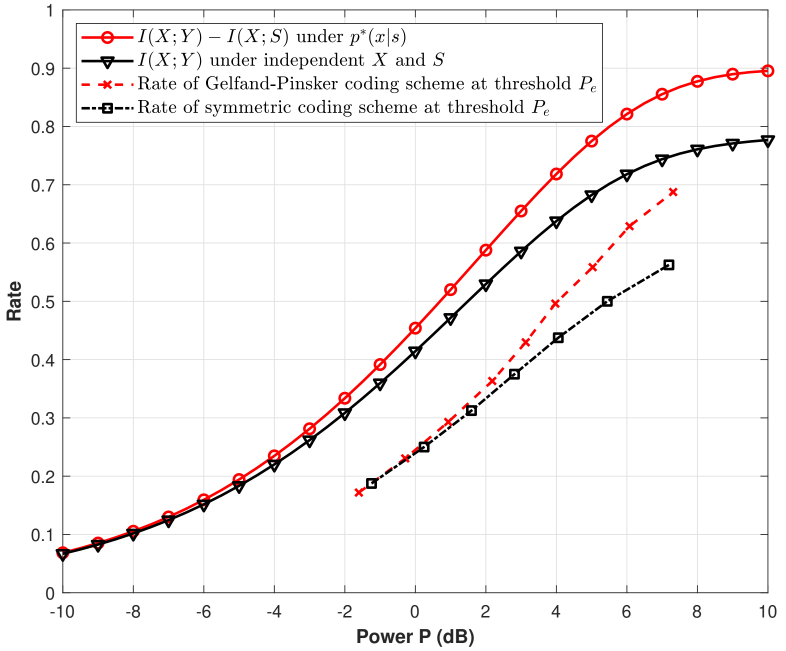


Figure 4.6: Achieved rates of the Gelfand–Pinsker coding scheme over a Gaussian channel with state at a fixed block error probability $P_e^{threshold} = 10^{-2}$.

successive cancellation decoding. The polar code is designed for the average channel $p_{\text{avg}}(y\,|\,x) \triangleq \sum_s p(s)p(y\,|\,x,s)$. Note that this symmetric coding scheme is a special case of our proposed Gelfand–Pinsker coding scheme for the case when $R_2 = 0$. Therefore, our goal here is to show that using strictly positive rates R_2 to target the conditional pmf $p^*(x\,|\,s)$ can achieve superior performance compared to the naive approach of coding for the average channel p_{avg}.

Let us first look at the theoretical limit of the two coding strategies as a function of power level P. As mentioned before, the maximum achievable rate of the proposed Gelfand–Pinsker coding scheme is $C_{\text{GP}} \triangleq I(X;Y) - I(X;S)$ evaluated under the maximizing conditional distribution $p^*(x\,|\,s)$, whereas the maximum achievable rate of the symmetric coding strategy is $C_{\text{sym}} \triangleq I(X;Y)$ when X is Bern(1/2) and independent of S. The plots with solid lines in Figure 4.6 show the evaluation of C_{GP} and C_{sym} as a function of P. The plots demonstrate that, in theory, Gelfand–Pinsker coding can achieve larger rates asymptotically compared to the symmetric coding strategy.

Before we explain the meaning of the dashed curves of Figure 4.6, we turn our attention to Figure 4.7, which plots the block error probability performance of the two coding strategies for the same block length $n = 1024$ and rate $R = 0.5$. To find the maximizing distribution $p^*(x\,|\,s)$ for the Gelfand–Pinsker coding scheme, our approach is to consider the power level P^* at which $C_{\text{GP}} = R+\gamma$. By choosing R_1 and R_2 according to (4.12), this guarantees that the overall rate of the Gelfand–Pinsker coding scheme is R. The power level P^* can be estimated from the plot of Figure 4.6.

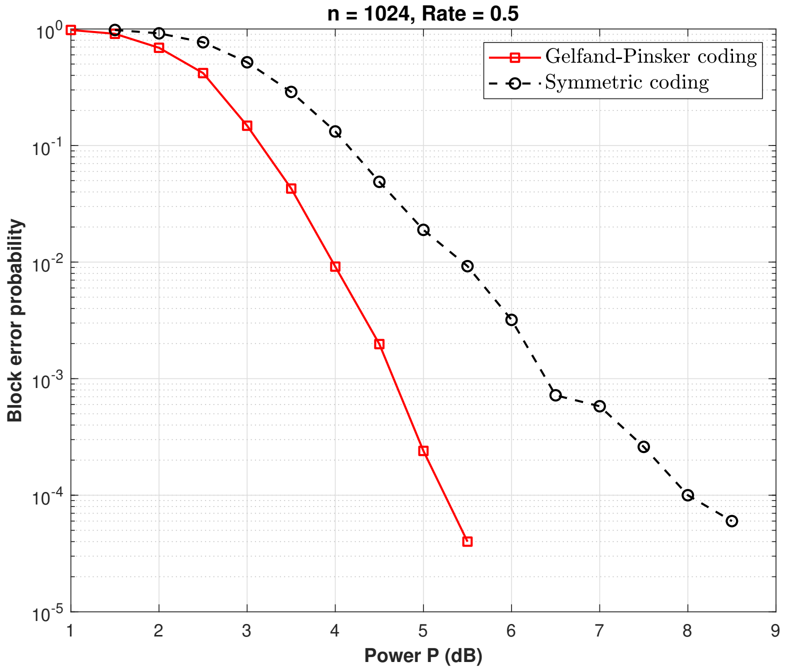


Figure 4.7: Simulation results of the Gelfand–Pinsker coding scheme for a block length $n = 1024$ and rate $R = 0.5$ over a Gaussian channel with state.

As shown in Figure 4.7, the Gelfand–Pinsker coding scheme shows significant performance gain compared to symmetric coding.

By repeating the same experiment for different values of the rate R, one can plot R as a function of the power level P, for a fixed block error probability $P_e^{\text{threshold}} = 10^{-2}$. This is shown in the dashed curves of Figure 4.6. For example, for a rate $R = 0.5$, we know from Figure 4.7 that the Gelfand–Pinkser coding scheme achieves an error probability of 10^{-2} when the power level is around 4 dB, whereas the symmetric coding scheme has an error probability of 10^{-2} when the power level is around 5.5 dB. A similar approach is taken for other values of R. The dashed curves of Figure 4.6 show that the Gelfand–Pinsker coding scheme can achieve larger rates in practice as well. These results demonstrate the importance of "shaping" the channel input with respect to the known state sequence. For more details about the simulation setup, our code is available on GitHub [82].

4.4.2 Marton Coding

Now, we verify the practicality of the Marton coding scheme through simulations. We consider a two-user Gaussian broadcast channel with two transmit antennas and a single receive antenna for each user. The channel can be modeled by the following input-output relations:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = H_{\text{ch}} W \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}, \tag{4.13}$$

where $H_{\text{ch}} = \begin{bmatrix} 1 & g \\ g & 1 \end{bmatrix}$ is the channel gain matrix with $g = 0.9$, W is a 2×2 precoding matrix used by the transmitter (if needed), $X_1 \in \{\pm 1\}$ and $X_2 \in \{\pm 1\}$ are BPSK-modulated signals corresponding to the channel input codewords (i.e., bit 0 is mapped to $+1$ and bit 1 is mapped to -1), $\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, I)$ is a vector of independent Gaussian noise components (I is the 2×2 identity matrix), and Y_1 and Y_2 are the channel outputs at the users' side. The transmitter wishes to send messages to the users at an overall *sum-rate* R_{sum}, while being subject to a sum-power constraint P such that

$$\mathsf{E}[\|WX_1^2\|^2] \leq P. \tag{4.14}$$

For a given input distribution $p(x_1, x_2)$ and a precoding matrix W satisfying the power constraint (4.14), recall from (4.5) that the sum-capacity can be expressed as

$$C_{\text{sum}}\big(p(x_1, x_2), W\big) \triangleq I(X_1; Y_1) + I(X_2; Y_2) - I(X_1; X_2). \tag{4.15}$$

The following coding strategies for the broadcast channel will be compared.

- "Marton coding with optimal precoding"[2]: corresponds to the proposed coding scheme that targets the channel input distribution $p^*_{\text{marton,opt}}(x_1, x_2)$ and precoding matrix $W^*_{\text{marton, opt}}$ that maximize the sum-capacity given in (4.15), while satisfying the power constraint (4.14).

- "Marton coding without precoding": corresponds to the proposed coding scheme that targets the channel input distribution $p^*_{\text{marton}}(x_1, x_2)$ that maximizes the sum-capacity given in (4.15), while setting the precoding matrix to $W = \sqrt{\frac{P}{2}} I$.

- "Symmetric coding with optimal precoding": corresponds to the strategy that sets the precoding matrix to $W^*_{\text{sym,opt}}$ that maximizes the sum-capacity given in (4.15), while setting the input distribution to the i.i.d. Bern(1/2) distribution (i.e., the two messages are encoded independently using two separate point-to-point channel codes, and the correlation between the two transmitted signals is attributed only to the linear precoder).

- "Symmetric coding without precoding": corresponds to the strategy that encodes the two messages independently using two separate point-to-point channel codes, while setting the precoding matrix to $W = \sqrt{\frac{P}{2}} I$.

[2] Marton coding over the Gaussian broadcast channel is often seen as an instance of "dirty paper coding" [71]. However, note that the channel input is binary in our case.

- "Minimum mean-square error (MMSE) precoding": corresponds to the strategy that encodes the two messages independently using two separate point-to-point channel codes, while applying the MMSE precoding matrix $W_{\text{MMSE}} = \lambda(H_{\text{ch}}^T H_{\text{ch}} + \frac{K}{P} I)^{-1} H_{\text{ch}}^T$ at the transmitter side, where $K = 2$ is the number of users and λ is a constant to satisfy the power constraint (4.14). The MMSE precoding matrix is sometimes referred to as the "transmit Wiener filter" [86].

- "Zero-forcing precoding": corresponds to the strategy that encodes the two messages independently using two separate point-to-point channel codes, while applying the zero-forcing precoding matrix $W_{\text{ZF}} = \lambda H_{\text{ch}}^{-1}$ at the transmitter side, where λ is a constant to satisfy the power constraint (4.14). Such a strategy suppresses the interference in the channel.

- "Time division": corresponds to the strategy that serves only a single user of the channel. That is, a single message is encoded using a point-to-point channel code, and the same codeword is transmitted across both antennas. The power allocation between the two transmit antennas is done so as to optimize the received signal-to-noise ratio. This is done by setting the precoding matrix to $W_{\text{time}-\text{division}} = \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{bmatrix}$, where (λ_1, λ_2) is the solution of

$$\begin{cases} \text{maximize} & (\sqrt{\lambda_1} + g\sqrt{\lambda_2})^2 \\ \text{subject to} & \lambda_1 + \lambda_2 = P, \\ & \lambda_1 \geq 0, \lambda_2 \geq 0. \end{cases}$$

Before simulating the proposed Marton coding scheme, we first plot the sum-capacity C_{sum} given in equation (4.15) for the different coding strategies. For the coding strategies that optimize over the channel input distribution and/or the precoding matrix, we use particle swarm optimization [87, 88] as an efficient heuristic method to perform the optimization. Figure 4.8 shows the plot of C_{sum} as a function of the sum-power constraint P for the different coding strategies over the two-user Gaussian broadcast channel model given by (4.13). Clearly, Marton coding with optimal precoding can achieve strictly larger sum-rates asymptotically compared to the other coding strategies. When no precoding is employed, Marton coding can still achieve significantly larger sum-rates compared to common linear precoding strategies used in practice such as MMSE precoding and zero-forcing precoding. This demonstrates the significance of stochastically shaping the channel input signals over the broadcast channel and motivates our Lego-brick design of a realizable Marton coding scheme using commercial off-the-shelf codes. Such an observation has been previously noted in the literature (e.g., [89, 90]).

Next, we simulate the different coding strategies over the Gaussian broadcast channel model using polar codes with successive cancellation decoding as the constituent point-to-point channel codes. The coding strategies are compared for the same block length $n = 1024$ and sum-rate $R_{\text{sum}} = 1$. Recall that each of the asymmetric channel code and the Gelfand–Pinkser code can

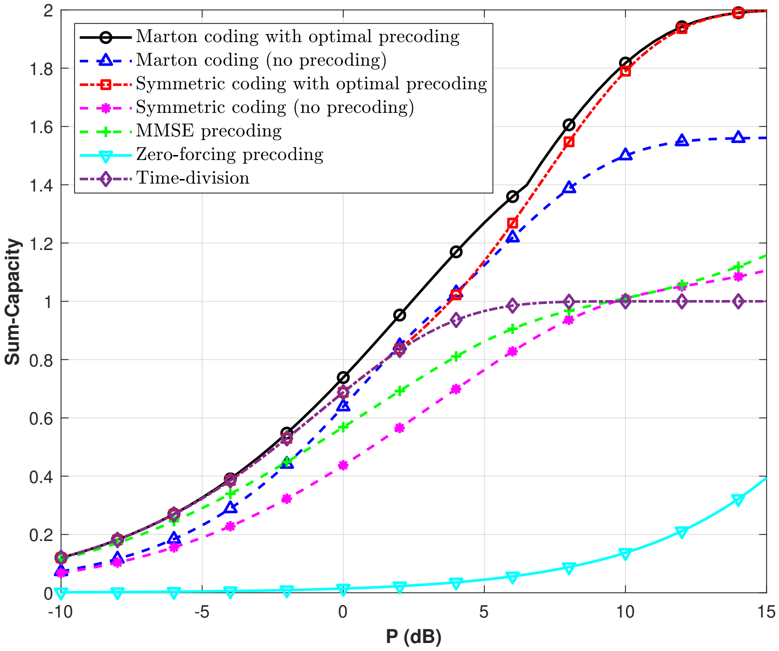


Figure 4.8: The sum-capacity for the different coding strategies over the broadcast channel.

be implemented using a pair of point-to-point channel codes, as described in Section 4.1. It follows that four polar codes are needed to implement the Marton coding scheme. Let $(R_{11}, R_{12}, R_{21}, R_{22})$ be the rates of the polar codes, where (R_{11}, R_{12}) are the rates of the two polar codes needed to construct the asymmetric channel code and (R_{21}, R_{22}) are the rates of the two polar codes needed to construct the Gelfand–Pinsker code. The rates $(R_{11}, R_{12}, R_{21}, R_{22})$ are chosen "close" to their theoretical limits (see Remarks 4.1.5 and 4.1.7). More precisely, for the Marton coding scheme, we take

$$\begin{aligned} R_{11} &= 1 - H(X_1 \,|\, Y_1) - \gamma, \\ R_{12} &= 1 - H(X_1), \\ R_{21} &= 1 - H(X_2 \,|\, Y_2) - \gamma, \\ R_{22} &= 1 - H(X_2 \,|\, X_1), \end{aligned} \tag{4.16}$$

where $\gamma > 0$ is a "back-off" parameter from the theoretical limit, which allows for a reasonable error probability[3]. Note that the sum-rate attained by the coding scheme is equal to $R_{11} - R_{12} + R_{21} - R_{22}$. Hence, in order to guarantee that this sum-rate is equal to R_{sum}, the target distribution $p^*_{\text{marton,opt}}(x_1, x_2)$ and precoding matrix $W^*_{\text{marton,opt}}$ are chosen to be the maximizers of C_{sum} when considering the power level P^* at which $C_{\text{sum}} = R_{\text{sum}} + 2\gamma$.[4] Given the target distribution

[3] Note that the back-off parameter is only used along R_{11} and R_{21} since these rates correspond to the polar codes used for error correction. In contrast, our simulations show that the polar codes used for shaping perform pretty well, even at rates very close to the theoretical limit; hence, no back-off is needed for R_{12} and R_{22}.

[4] The power level P^* can be estimated from the plot of Figure 4.8.

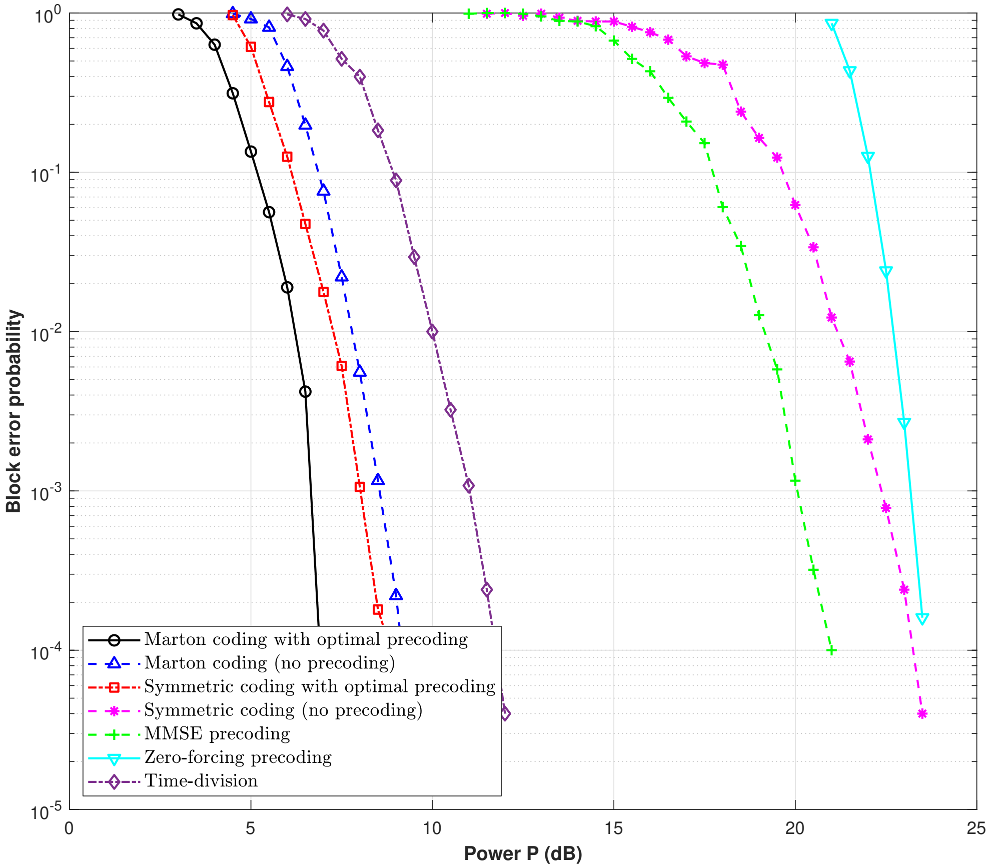


Figure 4.9: Simulation results for the different coding strategies over a two-user Gaussian broadcast channel for the same block length $n = 1024$ and sum-rate $R_{\text{sum}} = 1$.

$p^*_{\text{marton,opt}}(x_1, x_2)$ and the precoding matrix $W^*_{\text{marton,opt}}$, the rates $(R_{11}, R_{12}, R_{21}, R_{22})$ are chosen according to (4.16). A similar approach is taken to find the target distribution $p^*_{\text{marton}}(x_1, x_2)$ and the precoding matrix $W^*_{\text{sym,opt}}$. In our simulations over the broadcast channel, we used $\gamma = 1/16$. For more details about the simulation setup, our code is available on GitHub [82].

The simulation results are shown in Figure 4.9. Clearly, the Marton coding scheme with optimal precoding can achieve improved block error rate performance compared to the other coding strategies. Even when no precoding is used, the Marton coding scheme can achieve better performance compared to common coding strategies often employed in practice, such as time division, MMSE precoding and zero-forcing precoding.

Remark 4.4.1. Note that all the coding strategies that are considered in this part can be implemented as instances of the proposed Marton coding scheme (Figure 4.4) for particular choices of the rates $(R_{11}, R_{12}, R_{21}, R_{22})$, the target channel input distribution $p(x_1, x_2)$ and the precoding matrix W.

Acknowledgement

Chapter 4, in part, is a reprint with permission of the material as it appears in the papers: Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to coding for network communication,” arXiv:2211.07208, November 2022, which is in preparation to be submitted to *IEEE Transactions on Information Theory*, and Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to coding for asymmetric channels and channels with state,” in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 1367–1372, 2021. The dissertation author was the primary investigator and author of these papers. This work was supported in part by the Institute for Information & Communication Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-01659, 5G Open Intelligence-Defined RAN (ID-RAN) Technique based on 5G New Radio), in part by the NSERC Discovery Grant No. RGPIN-2019-05448, and in part by the NSERC Collaborative Research and Development Grant CRDPJ 543676-19.

Chapter 5

Coding over Cloud Radio Access Networks: A Lego-Brick Approach

In the last two chapters, we have constructed coding schemes for several multiterminal source and channel coding problems. In this chapter, we build on the previous constructions to address the problem of coding over multihop networks. In specific, we focus on cloud radio access networks.

Cloud radio access networks (C-RAN's) are a key part in the deployment of 5G systems [91, 92]. These networks model the case when multiple base stations are coordinated by a cloud-based central processor (CP) through capacity-limited wired or wireless links. In a downlink C-RAN scenario, the processing is done by the central processor assuming no capacity constraints, and then the baseband signals are digitized and sent over the capacity-limited links to the base stations. Similarly, in an uplink C-RAN scenario, the received signals at the based stations are digitized and sent over the capacity-limited links to the CP. Such an approach often calls for high link capacity requirements.

Alternatively, in this chapter, we consider coding schemes for the downlink and uplink C-RAN architecture (depicted in Figures 5.1 and 5.6 respectively) that view the entire system as a two-hop relay network. In this model, the base stations act as relays that send the prescribed signals from the CP to the users in the downlink scenario and compress the received signals to the CP in the uplink scenario. Such a model was studied, for example, in [93, 94, 95, 96, 97]. We construct coding schemes for both the downlink and uplink scenarios starting from basic coding blocks which we have previously considered in this dissertation. We show that the constructions can achieve well-known inner bounds for these network models. Moreover, through simulations, we demonstrate that the coding schemes can be implemented in real-world systems. We consider downlink C-RAN's in Section 5.1 and uplink C-RAN's in Section 5.2.

5.1 Downlink C-RAN

In this section, we present a coding scheme for the two-user, two-relay downlink C-RAN problem starting from a Marton code for a broadcast channel and two lossy source codes. The coding scheme can achieve the corner points of the generalized compression strategy characterized in [94]. First, let's start by defining the coding problem for the downlink C-RAN model.

5.1.1 Problem Statement

Consider the downlink of a cloud radio access network (C-RAN) with two users and two relays [42], as shown in Figure 5.1. A central processor (CP) communicates with the relays through noiseless fronthaul links of finite capacities C_1 and C_2. A memoryless channel $p(y_1, y_2 \,|\, x_1, x_2)$ is assumed between the relays and the users, with an input alphabet $\mathcal{X}^2 = \{0,1\}^2$ and output alphabet $\mathcal{Y}_1 \times \mathcal{Y}_2$. An (R_1, R_2, n) code for the downlink C-RAN problem consists of

- message sets $\mathcal{M}_1$, $\mathcal{M}_2$ such that $|\mathcal{M}_1| = 2^{nR_1}$ and $|\mathcal{M}_2| = 2^{nR_2}$,
- index sets $\mathcal{S}_1$, $\mathcal{S}_2$ such that $|\mathcal{S}_1| = 2^{nC_1}$ and $|\mathcal{S}_2| = 2^{nC_2}$,
- an encoder $g : \mathcal{M}_1 \times \mathcal{M}_2 \to \mathcal{S}_1 \times \mathcal{S}_2$ at the CP that maps each message pair (m_1, m_2) to a pair of indices $(s_1, s_2) = g(m_1, m_2)$,
- encoders $h_j : \mathcal{S}_j \to \mathcal{X}^n$ at the jth relay for $j = 1, 2$, that map each index s_j to a codeword $x_j^n = h_j(s_j)$,
- decoders $\psi_j : \mathcal{Y}_j^n \to \mathcal{M}_j$ for $j = 1, 2$, that assign message estimates $\hat{m}_j = \psi_1(y_j^n)$ to each received sequence y_j^n.

The average probability of error of the code is $\epsilon = \mathsf{P}\left\{\{\widehat{M}_1 \neq M_1\} \cup \{\widehat{M}_2 \neq M_2\}\right\}$. A rate pair (R_1, R_2) is said to be achievable for the downlink C-RAN problem if there exists a sequence of (R_1, R_2, n) codes with vanishing error probability asymptotically.

Coding schemes for the downlink C-RAN model have been proposed in [94, 93, 96, 97], among many others. In particular, Patil and Yu study in [94] a coding scheme for the downlink C-RAN problem which utilizes Marton's multicoding scheme for broadcasting followed by multivariate compression for transmission over the fronthaul links. The achievable rate region of Patil and Yu's coding scheme is the set of rate tuples (R_1, R_2) satisfying

$$\begin{aligned} R_1 &< I(U_1; Y_1), \\ R_2 &< I(U_2; Y_2), \\ R_1 + R_2 &< I(U_1; Y_1) + I(U_2; Y_2) - I(U_1; U_2), \end{aligned} \tag{5.1}$$

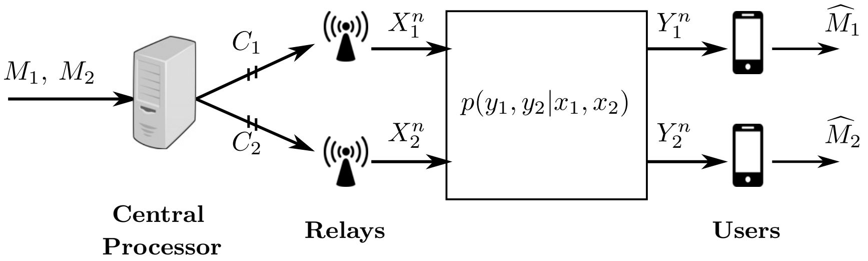


Figure 5.1: Downlink C-RAN problem with two users, two relays and a channel $p(y_1, y_2|x_1, x_2)$ between the relays and the users.

for some joint distribution $p(u_1, u_2, x_1, x_2)$ such that

$$\begin{aligned} C_1 &> I(U_1, U_2; X_1) \\ C_2 &> I(U_1, U_2; X_2) \\ C_1 + C_2 &> I(U_1, U_2; X_1, X_2) + I(X_1; X_2). \end{aligned} \tag{5.2}$$

Note that a strictly larger rate region can be achieved using the distributed decode-and-forward relaying scheme [93]. Notice the similarity of the rate region in (5.1) to the rate region of Marton coding over broadcast channels (equation (4.5)).

5.1.2 Coding Scheme

As mentioned before, the coding scheme for the downlink C-RAN model can be constructed starting from a Marton code for a broadcast channel and two lossy source codes. The Marton codes shape the channel input distribution while reliably sending messages to the intended users, whereas the lossy source codes compress the encoded signals to accommodate the limited capacity constraints of the fronthaul links. For illustrative purposes, we will describe the Marton code by its constituent asymmetric channel code and Gelfand–Pinsker code, as described in Section 4.2. Similarly, each lossy source code will be implemented using its constituent point-to-point channel code and lossless source code, as described in Section 3.2.3. Therefore, the following Lego bricks (one asymmetric channel code, one Gelfand–Pinsker code, two point-to-point channel codes and two lossless source codes) will be used in the construction. Inspired by Patil and Yu's random coding scheme, our coding scheme targets a joint distribution $p(u_1, u_2, x_1, x_2)$ at the CP[1]. Later, we show that the coding scheme can achieve a corner point in the Patil-Yu rate region given in (5.1) and a corner point in the compression region given in (5.2).

Lego Brick 5.1.1 (Asym → DL C-RAN): *an (R_1, n) asymmetric channel code (f_1, ψ_1) for the*

[1] For example, $p(u_1, u_2, x_1, x_2)$ can be chosen to maximize the maximum achievable sum-rate $I(U_1; Y_1) + I(U_2; Y_2) - I(U_1; U_2)$, while satisfying the compression constraints given in (5.2).

channel

$$p(y_1 | u_1) = \sum_{\substack{u_2, x_1, \\ x_2, y_2}} p(u_2, x_1, x_2 | u_1) p(y_1, y_2 | x_1, x_2),$$

which targets an input distribution $p(u_1)$*, such that, for* $M_1 \sim \mathrm{Unif}([2^{nR_1}])$*, the sequence* $U_1^n = g_1(M_1)$ *satisfies*

$$\frac{1}{2} \sum_{u_1^n} \left| P\{U_1^n = u_1^n\} - \prod_{i=1}^{n} p(u_{1i}) \right| \leq \delta_1, \tag{5.3}$$

for some $\delta_1 > 0$*. Let* ϵ_1 *be the average probability of error of the asymmetric channel code when the channel input distribution is i.i.d. according to* $p(u_1)$*.*

Lego Brick 5.1.2 (GP → DL C-RAN): *an* (R_2, n) *code* (f_2, ψ_2) *for the Gelfand–Pinsker problem defined by*

$$p(u_1) p(y_2 | u_2, u_1) = p(u_1) \sum_{x_1, x_2, y_1} p(x_1, x_2 | u_1, u_2) p(y_1, y_2 | x_1, x_2),$$

which targets a conditional distribution $p(u_2 | u_1)$*, such that, when* $M_2 \sim \mathrm{Unif}([2^{nR_2}])$ *and* $\widetilde{U}_1^n$ *is an i.i.d.* $p(u_1)$ *sequence, the sequence* $\widetilde{U}_2^n = g_2(M_2, \widetilde{U}_1^n)$ *satisfies*

$$\frac{1}{2} \sum_{u_1^n, u_2^n} \left| P\{\widetilde{U}_1^n = u_1^n, \widetilde{U}_2^n = u_2^n\} - \prod_{i=1}^{n} p(u_{1i}, u_{2i}) \right| \leq \delta_2, \tag{5.4}$$

for some $\delta_2 > 0$*. Let* ϵ_2 *be the average probability of error of the Gelfand–Pinsker code when the conditional distribution of the channel input given the channel state is i.i.d. according to* $p(u_2 | u_1)$*.*

Lego Brick 5.1.3 (P2P → Lossy → DL C-RAN): *a* (k_{31}, n) *linear point-to-point channel code* (H_{31}, ϕ_{31}) *with codebook* $\mathcal{C}_{31}$ *for the channel*

$$\bar{p}_3(u_1, u_2, v \,|\, x_1) = p_{U_1, U_2, X_1}(u_1, u_2, x_1 \oplus v). \tag{5.5}$$

Let δ_3 *denote the decoding distance of the code* (H_{31}, ϕ_{31}) *with respect to the channel* $\bar{p}_3$*.*

Lego Brick 5.1.4 (Lossless → Lossy → DL C-RAN): *an* $(n - k_{32}, n)$ *lossless source code* (H_{32}, ϕ_{32}) *for a* $\mathrm{Bern}(p_{X_1}(1))$ *source with codebook* $\mathcal{C}_{32}$ *and average probability of error* ϵ_3*. We further assume that* $\mathcal{C}_{32} \subseteq \mathcal{C}_{31}$ *(i.e., the two codes are nested) and that* $k_{31} - k_{32} < nC_1$*.*

Lego Brick 5.1.5 (P2P → Lossy → DL C-RAN): *a* (k_{41}, n) *linear point-to-point channel code* (H_{41}, ϕ_{41}) *with codebook* $\mathcal{C}_{41}$ *for the channel*

$$\bar{p}_4(u_1, u_2, x_1, v \,|\, x_2) = p_{U_1, U_2, X_1, X_2}(u_1, u_2, x_1, x_2 \oplus v). \tag{5.6}$$

Let δ_4 *denote the decoding distance of the code* (H_{41}, ϕ_{41}) *with respect to the channel* $\bar{p}_4$*.*

Lego Brick 5.1.6 (Lossless → Lossy → DL C-RAN): *an* $(n - k_{42}, n)$ *lossless source code* (H_{42}, ϕ_{42}) *for a* $\mathrm{Bern}(p_{X_2}(1))$ *source with codebook* $\mathcal{C}_{42}$ *and average probability of error* ϵ_4*. We further assume that* $\mathcal{C}_{42} \subseteq \mathcal{C}_{41}$ *(i.e., the two codes are nested) and that* $k_{41} - k_{42} < nC_2$*.*

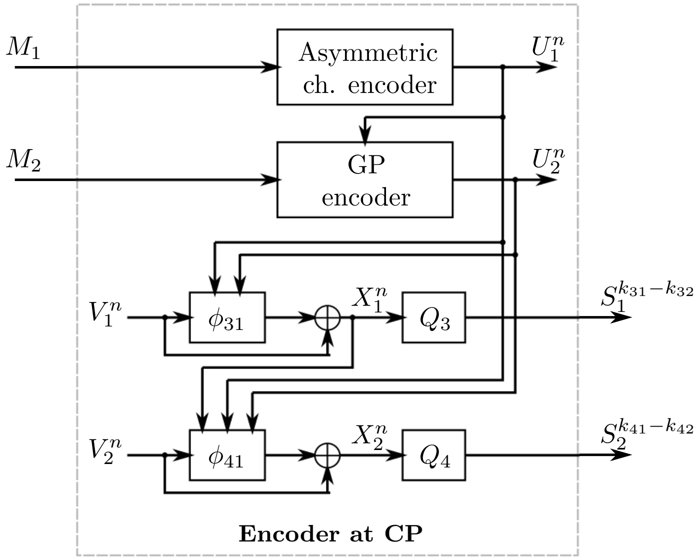


Figure 5.2: Encoding scheme at the central processor for the two-user, two-relay downlink C-RAN problem using an asymmetric channel code, a Gelfand–Pinsker code and two point-to-point channel codes.

Remark 5.1.1. The channels $\bar{p}_3$ and $\bar{p}_4$ are the symmetrized channels corresponding to the joint distributions $p\big(x_1,(u_1,u_2)\big)$ and $p\big(x_2,(u_1,u_2,x_1)\big)$, respectively.

Remark 5.1.2. Without loss of generality, assume that H_{j1} is a submatrix of H_{j2} for $j=3,4$, i.e., $H_{j2}=\begin{bmatrix} H_{j1} \\ Q_j \end{bmatrix}$ for some $(k_{j1}-k_{j2})\times n$ matrix Q_j.

Figure 5.2 and Figure 5.3 show the block diagrams of the coding scheme at the central processor, relays and users. The key point at the central processor is to construct a tuple $(U_1^n,U_2^n,X_1^n,X_2^n)$ that is close in total variation distance to the i.i.d. distribution according to $p(u_1,u_2,x_1,x_2)$, and then compress (X_1^n,X_2^n) through a pair of indices that are sent to the relays through the fronthaul links. The relays recover estimates of (X_1^n,X_2^n), which are transmitted to the users through the channel. Using standard arguments that were used throughout this dissertation, it follows from conditions (5.3), (5.4), and Lemma 3.2.1 that

$$\frac{1}{2}\sum_{\substack{u_1^n,u_2^n,\\ x_1^n,x_2^n}}\left|\mathsf{P}\{U_1^n=u_1^n,U_2^n=u_2^n,X_1^n=x_1^n,X_2^n=x_2^n\}-\prod_{i=1}^{n}p(u_{1i},u_{2i},x_{1i},x_{2i})\right|\leq\delta_1+\delta_2+\delta_3+\delta_4,$$

which says that the distribution of $(U_1^n,U_2^n,X_1^n,X_2^n)$ is $(\delta_1+\delta_2+\delta_3+\delta_4)$-away in total variation distance from the i.i.d. $p(u_1,u_2,x_1,x_2)$ distribution. Furthermore, the average probability of error of the coding scheme can be bounded as

$$\mathsf{P}\left\{\{\widehat{M}_1\neq M_1\}\cup\{\widehat{M}_2\neq M_2\}\right\}\;\leq\;\delta_1+\delta_2+\delta_3+\delta_4+\epsilon_1+\epsilon_2+\epsilon_3+\epsilon_4,$$

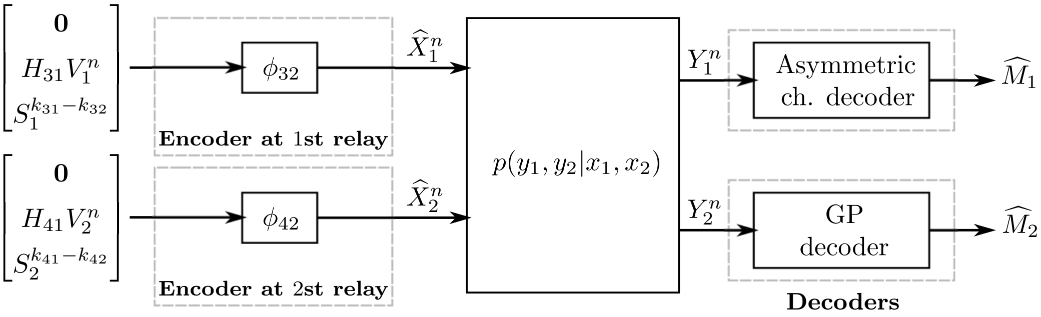


Figure 5.3: Encoding scheme at the relays and decoding scheme at the users for the downlink C-RAN problem.

following similar bounding techniques as in previous chapters.

The coding scheme attains the rate pair (R_1, R_2), and the compression rates are $\frac{k_{31}-k_{32}}{n}$ and $\frac{k_{41}-k_{42}}{n}$. Notice that these compression rates do not exceed the fronthaul link capacities (by assumption).

Remark 5.1.3. Similar to the analysis of the Marton coding scheme of Section 4.2, the rate tuple (R_1, R_2) can be made arbitrarily close to $(I(U_1; Y_1), I(U_2; Y_2) - I(U_2; U_1))$ for sufficiently large n, which is a corner point of the region of rate constraints, given in (5.1). On the other hand, following the discussion on the achievable rates of the lossy source coding scheme (specifically, Remarks 3.2.5 and 3.2.6), the compression rate pair $\left(\frac{k_{31}-k_{32}}{n}, \frac{k_{41}-k_{42}}{n}\right)$ can be made arbitrarily close to $\big(I(U_1, U_2; X_1), I(U_1, U_2, X_1; X_2)\big)$ for sufficiently large n, which is a corner point of the region of compression constraints, given in (5.2). It follows that our coding scheme can be used to achieve a corner point of the Patil-Yu region. If the encoding order is modified, other corner points of the rate region can be achieved.

Conclusion: By adapting a similar approach for the general K-user, L-relay downlink C-RAN problem, a coding scheme for this problem can be constructed starting from one asymmetric channel code, $K-1$ Gelfand–Pinsker codes, L point-to-point channel codes and L lossless source codes.

5.1.3 Simulation Results

By implementing each of the lossless source codes, the Gelfand–Pinsker code and the asymmetric channel code using their constituent point-to-point channel codes as described in Sections 3.1.4, 4.1.2 and 4.1.3, the coding scheme for downlink C-RAN can be simulated using point-to-point channel codes. We consider a two-user two-relay model with a Gaussian channel between the relays and the users, i.e., the channel output can be expressed as

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = H_{\text{ch}} \Lambda \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \tag{5.7}$$

where $H_{\text{ch}} = \begin{bmatrix} 1 & g \\ g & 1 \end{bmatrix}$ is the channel gain matrix with $g = 0.9$, $\Lambda = \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{bmatrix}$ is a power allocation matrix for the relays[2], $X_1 \in \{\pm 1\}$ and $X_2 \in \{\pm 1\}$ are BPSK-modulated signals, and $\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \sim \mathcal{N}\big(\mathbf{0}, I\big)$ is a vector of independent Gaussian noise components. The transmitter wishes to send messages to the users at an overall sum-rate R_{sum}, and each relay is subject to a power constraint P such that $\lambda_j \leq P$ for each $j = 1, 2$.

We first look at the sum-capacity of the proposed coding scheme under different power constraints P and fronthaul capacity constraints C_1 and C_2. We know from (5.1) that the sum-capacity of the coding scheme is given by

$$C_{\text{sum}} \triangleq \max_{\substack{p(u_1,u_2,x_1,x_2) \in \mathcal{D}(C_1,C_2), \\ (\lambda_1,\lambda_2) \in \mathcal{K}(P)}} I(U_1;Y_1) + I(U_2;Y_2) - I(U_1;U_2), \tag{5.8}$$

where $\mathcal{D}(C_1, C_2)$ is the set of joint distributions $p(u_1, u_2, x_1, x_2)$ that satisfy the compression constraints given in (5.2), and

$$\mathcal{K}(P) = \{(\lambda_1, \lambda_2) : 0 \leq \lambda_1 \leq P, 0 \leq \lambda_2 \leq P\}.$$

We use the genetic algorithm [98, 88] as an efficient heuristic method to perform the optimization in (5.8). Figure 5.4 shows the plot of C_{sum} as a function of the power constraint P under different fronthaul capacity constraints C_1 and C_2. As expected, the sum-capacity of the proposed coding scheme is improved when the fronthaul link capacities are increased. Note that when $C_1 \geq 1$ and $C_2 \geq 1$, the sum-capacity of the downlink C-RAN coding scheme corresponds to that of the Marton coding scheme of Section 4.4.2 when the transmitter is subject to a per-antenna power constraint and the precoding matrix is restricted to be a diagonal matrix.

Next, the proposed coding scheme for the downlink C-RAN problem is simulated for a block length $n = 1024$ and a sum-rate $R_{\text{sum}} = 0.75$ using polar codes with successive cancellation decoding as the constituent point-to-point channel codes. Note that each of the asymmetric channel code and the Gelfand–Pinsker code can be implemented using a pair of point-to-point channel codes, and each of the lossless source codes can be implemented using a single point-to-point channel code. Hence, the downlink C-RAN coding scheme can be constructed starting from eight polar codes. Let $(R_{11}, R_{12}, R_{21}, R_{22}, R_{31}, R_{32}, R_{41}, R_{42})$ be the rates of the constituent polar codes, where (R_{11}, R_{12}) are the rates of the two polar codes needed to construct the asymmetric channel code, (R_{21}, R_{22})

[2] Note that since the two relays do not communicate and the fronthaul links are capacity-limited, we do not consider the possibility of applying a general precoding matrix as in Section 4.4.2.

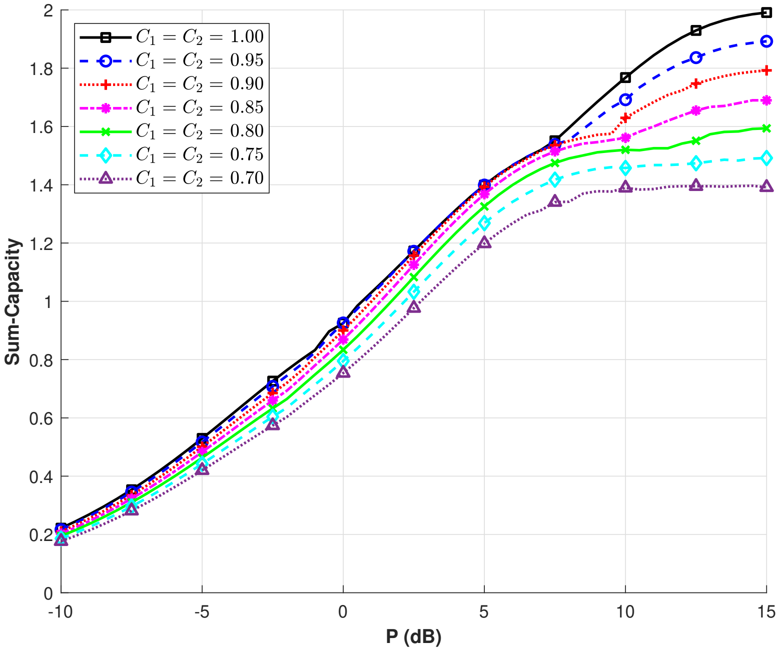


Figure 5.4: The sum-capacity of the proposed coding scheme for the downlink C-RAN problem under different fronthaul capacity constraints.

are the rates of the two polar codes needed to construct the Gelfand–Pinsker code, (R_{31}, R_{32}) are the rates of the two polar codes needed to construct the codes $\mathcal{C}_{31}$ and $\mathcal{C}_{32}$, and (R_{41}, R_{42}) are the rates of the two polar codes needed to construct the codes $\mathcal{C}_{41}$ and $\mathcal{C}_{42}$. The rates are chosen "close" to their theoretical limits, i.e., we set

$$\begin{aligned}
&R_{11} = 1 - H(U_1 \,|\, Y_1) - \gamma_r, && R_{31} = 1 - H(X_1 \,|\, U_1, U_2),\\
&R_{12} = 1 - H(U_1), && R_{32} = 1 - H(X_1) - \gamma_c,\\
&R_{21} = 1 - H(U_2 \,|\, Y_2) - \gamma_r, && R_{41} = 1 - H(X_2 \,|\, U_1, U_2, X_1),\\
&R_{22} = 1 - H(U_2 \,|\, U_1), && R_{42} = 1 - H(X_2) - \gamma_c,
\end{aligned} \tag{5.9}$$

where $\gamma_r > 0$ and γ_c are "back-off" parameters from the theoretical limit that are used for the polar codes involved in error correction (i.e., not shaping). Note that the sum-rate attained by the coding scheme is equal to $R_{11} - R_{12} + R_{21} - R_{22}$. Hence, in order to guarantee that this sum-rate is equal to R_{sum}, the coding scheme targets the joint distribution $p^*(u_1, u_2, x_1, x_2)$ and the power levels $(\lambda_1^*, \lambda_2^*)$ that maximize C_{sum} under the optimization problem of (5.8), where the power constraint P is set to be equal to the power level P^* at which $C_{\text{sum}} = R_{\text{sum}} + 2\gamma_r$.[3] Under the joint distribution $p^*(u_1, u_2, x_1, x_2)$ and the power levels $(\lambda_1^*, \lambda_2^*)$, the rates of the constituent polar codes are set according to (5.9). Note that γ_c should be chosen so that $R_{31} - R_{32} \leq C_1$ and $R_{41} - R_{42} \leq C_2$.

[3] Note that the power level P^* can be estimated using the plot of Figure 5.4.

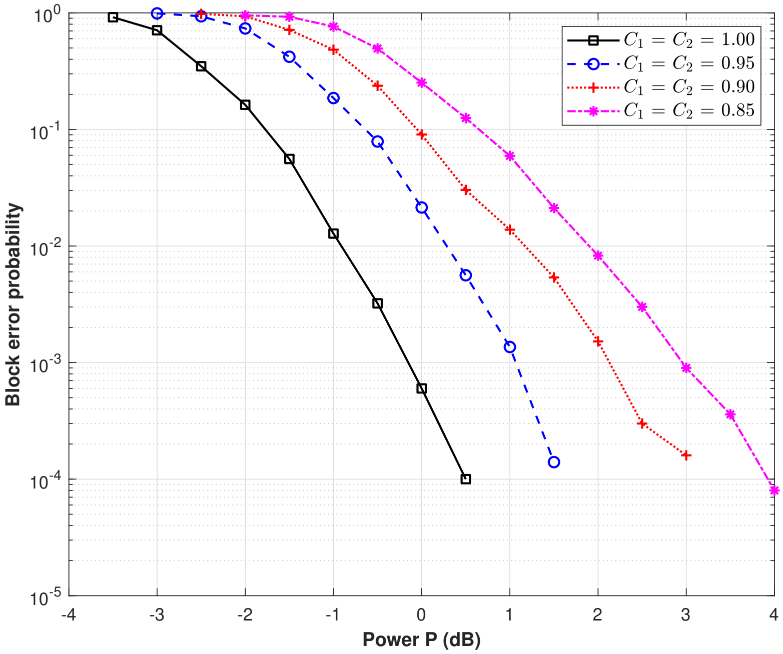


Figure 5.5: Simulation results of the downlink C-RAN coding scheme for a block length $n = 1024$ and sum-rate $R_{\text{sum}} = 0.75$ under different fronthaul capacity constraints.

In our simulations over the downlink C-RAN channel model, we used $\gamma_r = 1/8$ and $\gamma_c = 5/32$. For more details about the simulation setup, our code is available on GitHub [82]. The block error rate performance of the downlink C-RAN coding scheme is shown in Figure 5.5 for different fronthaul capacity constraints C_1 and C_2. The results demonstrate the practicality of the proposed coding scheme over the downlink C-RAN channel model.

5.2 Uplink C-RAN

Now, we turn our attention to the uplink C-RAN channel model, as depicted in Figure 5.6. We construct a coding scheme for this model starting from a multiple access channel code and a Berger–Tung code. We show that the coding scheme can achieve a corner point of the network compress-and-forward rate region with successive decoding characterized in [95]. As usual, we start with defining the coding problem for uplink C-RAN's.

5.2.1 Problem Statement

Consider the uplink of a cloud radio access network (C-RAN) with two users and two relays [42], as shown in Figure 5.6. Two users wish to communicate with a central processor (CP) through two relays that are connected to the CP through noiseless backhaul links of finite capacities

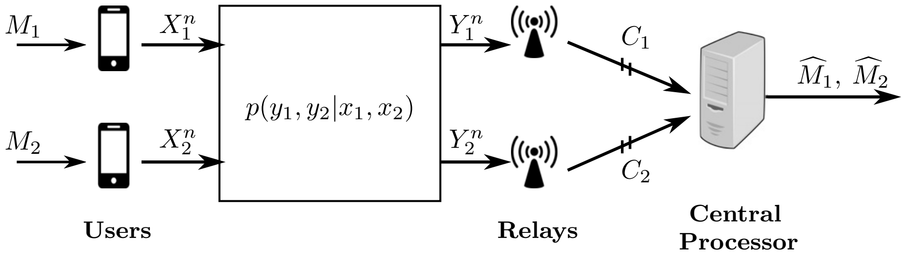


Figure 5.6: Uplink C-RAN problem with two users and two relays.

C_1 and C_2. A memoryless channel $p(y_1, y_2 \,|\, x_1, x_2)$ is assumed between the users and the relays, with an input alphabet $\mathcal{X}^2 = \{0,1\}^2$ and output alphabet $\mathcal{Y}_1 \times \mathcal{Y}_2$. An (R_1, R_2, n) code for uplink C-RAN problem consists of

- message sets $\mathcal{M}_1$, $\mathcal{M}_2$ such that $|\mathcal{M}_1| = 2^{nR_1}$ and $|\mathcal{M}_2| = 2^{nR_2}$,
- index sets $\mathcal{S}_1$, $\mathcal{S}_2$ such that $|\mathcal{S}_1| = 2^{nC_1}$ and $|\mathcal{S}_2| = 2^{nC_2}$,
- encoders $g_j : \mathcal{M}_j \to \mathcal{X}^n$ at the jth user for $j = 1, 2$ that map each message m_j to a codeword x_j^n,
- encoders $h_j : \mathcal{Y}_j^n \to \mathcal{S}_j$ at the jth relay for $j = 1, 2$ that map each received sequence y_j^n to an index $s_j = h_j(y_j^n)$,
- decoder $\psi : \mathcal{S}_1 \times \mathcal{S}_2 \to \mathcal{M}_1 \times \mathcal{M}_2$ at the CP that assign message estimates $(\hat{m}_1, \hat{m}_2)$ to each index pair (s_1, s_2).

The average probability of error of the code is defined as $\epsilon = \mathsf{P}\big\{\{\widehat{M}_1 \neq M_1\} \cup \{\widehat{M}_2 \neq M_2\}\big\}$. A rate pair (R_1, R_2) is said to be achievable for the uplink C-RAN problem if there exists a sequence of (R_1, R_2, n) codes with vanishing error probability asymptotically.

Coding schemes for the uplink C-RAN problem have been proposed in [99, 100, 93] based on the network compress-and-forward relaying scheme [101]. In [95], it is shown that for a particular decoding order of messages and quantization codewords at the CP, successive decoding can achieve the same maximum sum-rate as the joint decoding approach of the coding schemes in [99, 100, 93]. In particular, the achievable rate region of network compress-and-forward with successive decoding is the closure of the convex hull of all rate pairs (R_1, R_2) satisfying [95]

$$\begin{aligned} R_1 &< I(X_1; \widehat{Y}_1, \widehat{Y}_2 \,|\, X_2), \\ R_2 &< I(X_2; \widehat{Y}_1, \widehat{Y}_2 \,|\, X_1), \\ R_1 + R_2 &< I(X_1, X_2; \widehat{Y}_1, \widehat{Y}_2), \end{aligned} \tag{5.10}$$

for some product distribution $p(x_1)p(x_2)p(\hat{y}_1 \,|\, y_1)p(\hat{y}_2 \,|\, y_2)$ such that

$$\begin{aligned} C_1 &> I(Y_1; \widehat{Y}_1 \,|\, \widehat{Y}_2), \\ C_2 &> I(Y_2; \widehat{Y}_2 \,|\, \widehat{Y}_1), \\ C_1 + C_2 &> I(Y_1, Y_2; \widehat{Y}_1, \widehat{Y}_2). \end{aligned} \tag{5.11}$$

Notice the similarity of the rate region in (5.10) to the rate region of a multiple access channel (equation (4.8)), and the similarity of the compression constraints in (5.11) to the achievable rate region by the Berger–Tung coding scheme for the distributed lossy compression problem (equation (3.5)).

5.2.2 Coding Scheme

Now, we construct a coding scheme for the uplink C-RAN problem starting from a multiple access channel code and a Berger–Tung code. The coding scheme will target a given product distribution $p(x_1)p(x_2)p(\hat{y}_1 \,|\, y_1)p(\hat{y}_2 \,|\, y_2)$[4]. Therefore, the desired input-output joint distribution can be given by

$$p(x_1, x_2, y_1, y_2, \hat{y}_1, \hat{y}_2) = p(x_1)p(x_2)p(y_1, y_2 \,|\, x_1, x_2)p(\hat{y}_1 \,|\, y_1)p(\hat{y}_2 \,|\, y_2).$$

The multiple access channel code will encode the messages over the channel $p(\hat{y}_1, \hat{y}_2 \,|\, x_1, x_2)$, whereas the Berger–Tung will compress the received channel outputs at the relays with a joint distribution $p(y_1, y_2)$. More specifically, the coding scheme for the uplink C-RAN problem uses the following Lego bricks.

Lego Brick 5.2.1 (MAC → UL C-RAN): *an (R_1, R_2, n) code (f_1, f_2, ϕ) for the multiple access channel $p(\hat{y}_1, \hat{y}_2 \,|\, x_1, x_2)$ which targets input distributions $p(x_1)p(x_2)$, such that, for $M_j \sim \mathrm{Unif}([2^{nR_j}])$, the channel input $X_j^n = f_j(M_j)$ satisfies*

$$\frac{1}{2} \sum_{x_j^n} \left| P\{X_j^n = x_j^n\} - \prod_{i=1}^{n} p(x_{ji}) \right| \le \delta_j^{\mathrm{MAC}}, \tag{5.12}$$

for $j = 1, 2$ and some δ_1^{MAC}, $\delta_2^{\mathrm{MAC}} > 0$. Let ϵ be the average probability of error of the multiple access channel code when the channel input distributions are i.i.d. according to $p(x_1)p(x_2)$.

Lego Brick 5.2.2 (BT → UL C-RAN): *an (R_3, R_4, n) Berger–Tung code (f_3, f_4, ψ) for a $p(y_1, y_2)$-source that targets conditional distributions $p(\hat{y}_1 \,|\, y_1)p(\hat{y}_2 \,|\, y_2)$, such that, for $(Y_1^n, Y_2^n) \overset{\mathrm{iid}}{\sim} p(y_1, y_2)$, the sequences $(\widehat{Y}_1^n, \widehat{Y}_2^n) = \psi\big(g_3(Y_1^n), g_4(Y_2^n)\big)$ satisfy*

$$\frac{1}{2} \sum_{y_j^n,\, \hat{y}_j^n} \left| P\{Y_j^n = y_j^n, \widehat{Y}_j^n = \hat{y}_j^n\} - \prod_{i=1}^{n} p(y_{ji})p(\hat{y}_{ji} \,|\, y_{ji}) \right| \le \delta_j^{\mathrm{BT}} \tag{5.13}$$

for $j = 1, 2$ and some δ_1^{BT}, $\delta_2^{\mathrm{BT}} > 0$. Furthermore, we assume that $R_3 < C_1$ and $R_4 < C_2$.

[4] For example, such a product distribution can be chosen to maximize the maximum achievable sum-rate $I(X_1, X_2; \widehat{Y}_1, \widehat{Y}_2)$.

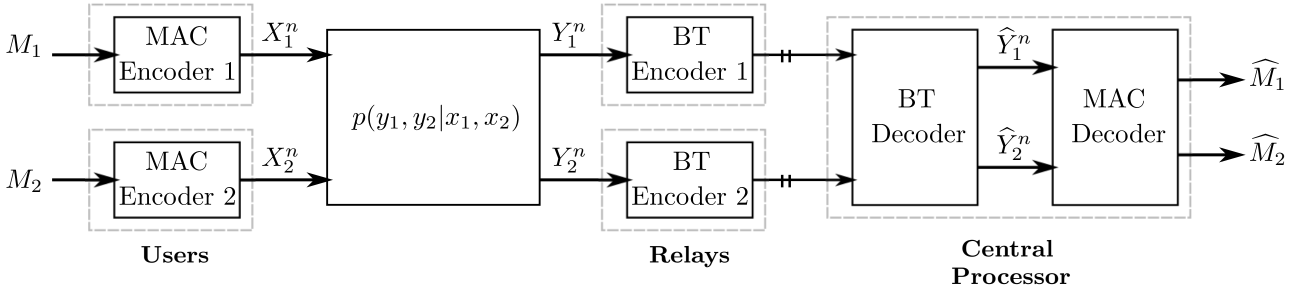


Figure 5.7: Coding scheme for the uplink C-RAN problem using a multiple access channel code and a Berger–Tung code.

Remark 5.2.1. The multiple access channel code can be implemented using two asymmetric channel codes, as described in Section 4.3, and the Berger–Tung code can be implemented using a lossy source code and a Wyner–Ziv code, as described in Section 3.4.

Figure 5.7 shows the block diagram of the uplink C-RAN problem. The multiple access channel code is used to encode the user messages while shaping the channel input signals, whereas the Berger–Tung code is used to compress the received signals over the backhaul links. The central processor first decodes the compressed codewords using the Beger–Tung decoder, then decodes the user messages using the multiple access channel decoder. Following similar bounding techniques seen thus far, the average probability of error of the coding scheme can be bounded as

$$\mathsf{P}\left\{\{\widehat{M}_1 \neq M_1\} \cup \{\widehat{M}_2 \neq M_2\}\right\} \leq \epsilon + \delta_1^{\mathrm{MAC}} + \delta_2^{\mathrm{MAC}} + \delta_1^{\mathrm{BT}} + \delta_1^{\mathrm{BT}}.$$

The user rate pair is given by (R_1, R_2). Note that the conditions $R_3 < C_1$ and $R_4 < C_2$ are needed so that the compression rates of the coding scheme do not exceed the backhaul link capacities.

Remark 5.2.2. For sufficiently large n, the user rate pair (R_1, R_2) can be chosen to be arbitrarily close to a corner point of the rate region given in (5.10). Similarly, for sufficiently large n, the compression rate pair (R_3, R_4) can be chosen to be arbitrarily close to a corner point of the compression rate region given in (5.11).

Remark 5.2.3. Note that the uplink C-RAN coding scheme can be implemented using point-to-point channel codes by constructing the multiple access channel and Berger–Tung codes using their constituent point-to-point channel codes, as described in Sections 4.3 and 3.4 respectively. Nevertheless, note that our construction of a Berger–Tung code using point-to-point channel codes requires binary sources. Hence, the channel output alphabets $\mathcal{Y}_1$ and $\mathcal{Y}_2$ of the uplink C-RAN model should be binary in this case.

Conclusion: By adapting the coding scheme to the general K-user, L-relay uplink C-RAN problem, a coding scheme in the general case can be constructed starting from a K-user multiple access channel and a Berger–Tung code for a distributed lossy compression problem with L sources.

5.2.3 Simulation Results

By implementing the multiple access channel code and the Berger–Tung code using their constituent point-to-point channel codes as described in Sections 4.3 and 3.4 respectively, the coding scheme for uplink C-RAN can be simulated using point-to-point channel codes. We consider a two-user two-relay model with a binary-quantized Gaussian channel between the users and the relays. That is, the channel output at the relays can be expressed as

$$\begin{aligned} \widetilde{Y}_1 &= X_1 + gX_2 + Z_1, \\ \widetilde{Y}_2 &= X_2 + gX_1 + Z_2, \\ Y_1 &= q_1(\widetilde{Y}_1), \\ Y_2 &= q_2(\widetilde{Y}_2), \end{aligned}$$

where $g = 0.9$, $(X_1, X_2) \in \{-\sqrt{P}, \sqrt{P}\}^2$ are BPSK-modulated signals, $(Z_1, Z_2) \sim \mathcal{N}(\mathbf{0}, I)$ are independent Gaussian noise components, and $q_1(.)$ and $q_2(.)$ are binary quantizers. The use of binary quantizers is motivated by the cases when the backhaul links are severely capacity-limited (e.g., when $C_1 \leq 1$ and $C_2 \leq 1$).

We first look at the sum-capacity of the proposed coding scheme as a function of the power constraint P and the backhaul capacity constraints C_1 and C_2. We know from (5.10) that the sum-capacity of the coding scheme can be expressed as

$$C_{\text{sum}} \triangleq \max_{p(x_1)p(x_2)p(\hat{y}_1|y_1)p(\hat{y}_2|y_2) \in \mathcal{D}(C_1, C_2)} I(X_1, X_2; \widehat{Y}_1, \widehat{Y}_2), \tag{5.14}$$

where $\mathcal{D}(C_1, C_2)$ is the set of all product distributions $p(x_1)p(x_2)p(\hat{y}_1 | y_1)p(\hat{y}_2 | y_2)$ that satisfy the compression constraints given in (5.11). We use the genetic algorithm [98, 88] as an efficient heuristic method to perform the optimization in (5.14). The quantizers $q_1(.)$ and $q_2(.)$ are optimized using the Lloyd-Max algorithm [102, 103]. That is, for any pair of input distributions $p(x_1)$ and $p(x_2)$, the distribution of $\widetilde{Y}_1$ (resp., $\widetilde{Y}_2$) is derived and used to compute the optimal partition points and quantization levels of $q_1(.)$ (resp., $q_2(.)$) using Lloyd-Max. Fig. 5.8 shows the plot of C_{sum} as a function of the power constraint P under different backhaul capacity constraints C_1 and C_2.

Next, the proposed coding scheme for the uplink C-RAN problem is simulated for a block length $n = 1024$ and a sum-rate $R_{\text{sum}} = 0.25$ using polar codes with successive cancellation decoding as the constituent point-to-point channel codes. Note that the multiple access channel code can be implemented using two asymmetric channel codes, each of which can be implemented using a pair of point-to-point channel codes. Moreover, the Berger–Tung code can be implemented using a lossy source code and a Wyner–Ziv code, each of which can be implemented using a pair of point-to-point channel codes. Hence, the uplink C-RAN coding scheme can be constructed starting from eight polar codes. Let $(R_{11}, R_{12}, R_{21}, R_{22}, R_{31}, R_{32}, R_{41}, R_{42})$ be the rates of the constituent polar codes, where (R_{11}, R_{12}) and (R_{21}, R_{22}) are the rates of the two polar codes needed to construct the first

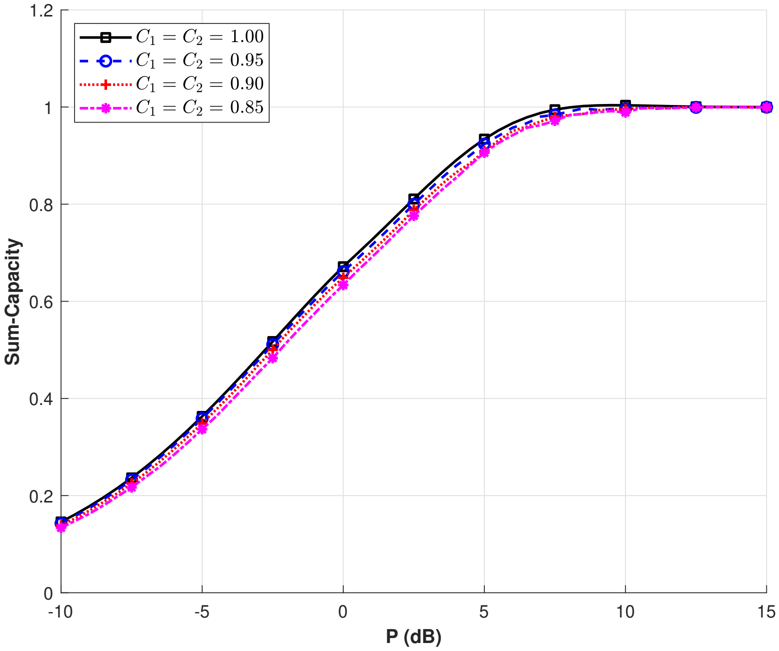


Figure 5.8: The sum-capacity of the proposed coding scheme for the uplink C-RAN problem under different backhaul capacity constraints.

and second asymmetric channel code respectively, and (R_{31}, R_{32}) and (R_{41}, R_{42}) are the rates of the two polar codes needed to construct the lossy source code and Wyner–Ziv code respectively. The rates are chosen "close" to their theoretical limits, i.e., we set[5]

$$\begin{aligned} R_{11} &= 1 - H(X_1 \,|\, \widehat{Y}_1, \widehat{Y}_2) - \gamma_r, & R_{31} &= 1 - H(\widehat{Y}_1 \,|\, Y_1), \\ R_{12} &= 1 - H(X_1), & R_{32} &= 1 - H(\widehat{Y}_1) - \gamma_c, \\ R_{21} &= 1 - H(X_2 \,|\, \widehat{Y}_1, \widehat{Y}_2, X_1) - \gamma_r, & R_{41} &= 1 - H(\widehat{Y}_2 \,|\, Y_2), \\ R_{22} &= 1 - H(X_2), & R_{42} &= 1 - H(\widehat{Y}_2 \,|\, \widehat{Y}_1) - \gamma_c, \end{aligned} \tag{5.15}$$

where $\gamma_r > 0$ and γ_c are "back-off" parameters from the theoretical limit that are used for the polar codes involved in error correction (i.e., not shaping). Note that the sum-rate attained by the coding scheme is equal to $R_{11} - R_{12} + R_{21} - R_{22}$. Hence, in order to guarantee that this sum-rate is equal to R_{sum}, the coding scheme targets the distributions $p(x_1)p(x_2)p(\hat{y}_1 \,|\, y_1)p(\hat{y}_2 \,|\, y_2)$ that maximize C_{sum} under the optimization problem of (5.14), where the power constraint P is set to be equal to the power level P^* at which $C_{\text{sum}} = R_{\text{sum}} + 2\gamma_r$.[6] Under the target distributions, the rates of the constituent polar codes are set according to (5.15). Note that γ_c should be chosen so that $R_{31} - R_{32} \leq C_1$ and $R_{41} - R_{42} \leq C_2$. In our simulations over the uplink C-RAN channel

[5] These theoretical limits can be derived for the uplink C-RAN model as a consequence of Remarks 3.2.6, 3.3.1, and 4.1.7.

[6] Note that the power level P^* can be estimated using the plot of Fig. 5.8.

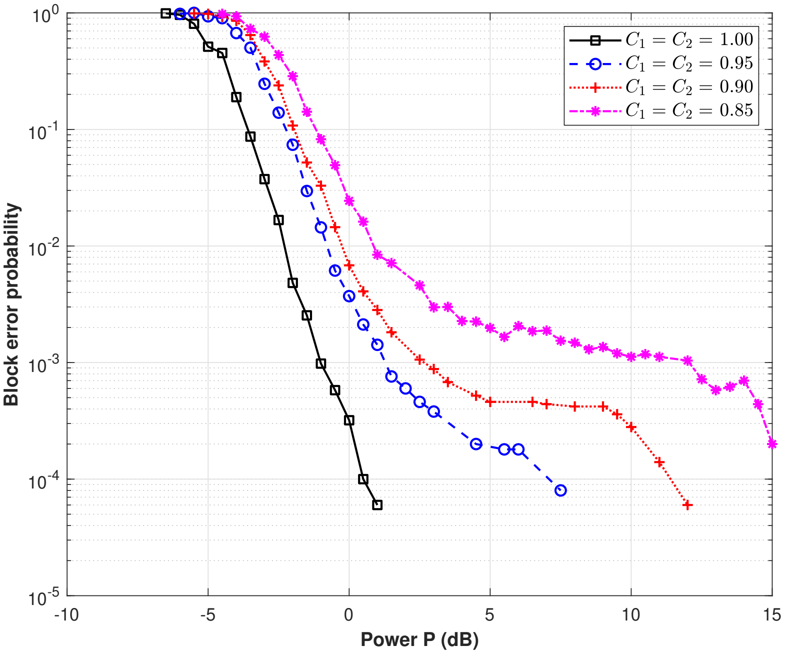


Figure 5.9: Simulation results of the uplink C-RAN coding scheme for a block length $n = 1024$ and sum-rate $R_{\text{sum}} = 0.25$ under different backhaul capacity constraints.

model, we used $\gamma_r = 1/8$ and $\gamma_c = 5/32$. For more details about the simulation setup, our code is available on GitHub [82]. The block error rate performance of the uplink C-RAN coding scheme is shown in Fig. 5.9 for different backhaul capacity constraints C_1 and C_2. The results demonstrate the practicality of the proposed coding scheme over the uplink C-RAN channel model. Note that the observed error floor is due to binary quantization.

Acknowledgement

Chapter 5, in part, is a reprint with permission of the material as it appears in the paper: Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to coding for network communication,” arXiv:2211.07208, November 2022, which is in preparation to be submitted to *IEEE Transactions on Information Theory*. The dissertation author was the primary investigator and author of these papers. This work was supported in part by the Institute for Information & Communication Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-01659, 5G Open Intelligence-Defined RAN (ID-RAN) Technique based on 5G New Radio), in part by the NSERC Discovery Grant No. RGPIN-2019-05448, and in part by the NSERC Collaborative Research and Development Grant CRDPJ 543676-19.

Chapter 6

Block-Markov Coding over Networks: A Lego-Brick Approach

6.1 Motivation

In most constructions we presented so far in this dissertation, the decoding distance of a point-to-point channel code designed for a BMS channel was used as a primitive property in one of the constituent Lego bricks. Unfortunately, the decoding distance is difficult to be estimated for commercial off-the-shelf codes since it involves the computation of the total variation distance between distributions over exponentially large alphabets. To circumvent this inconvenience, we present in this chapter modified coding schemes involving properties that are easily verifiable in practice. More precisely, the assumptions that we will be made on the Lego bricks in this chapter will be over alphabets with size that is at most polynomial in the block length. Moreover, the coding schemes presented here will not make any assumption of nestedness between the constituent Lego bricks, which was often assumed in the constructions thus far. However, as we shall see, these benefits will come at the cost of larger implementation complexity, a small penalty incurred in achievable rates, and a worse performance guarantee. In particular, the coding schemes presented in this section will have a *block-Markov structure*, i.e., they will be defined upon the transmission of several blocks of information, and the inputs to one encoding/decoding block can depend on the outputs of previous blocks.

The modified coding schemes presented in this chapter are based on assumptions made about the distributions of the decoding Hamming distance spectrum or the decoding joint-type spectrum (both terms will be defined shortly). The assumptions made are easily verifiable in practice since the both distributions can be estimated efficiently for any commercial off-the-shelf code via Monte Carlo simulations. It turns out that all the previous constructions can be modified to work under these simplified assumptions. Moreover, the block-Markov constructions will not require any constituent codebooks to be nested, unlike the previous constructions which often assumed that the codebook corresponding to one Lego brick is a subcode of another. One major consequence of these constructions is the following: one can assemble *any* set of point-to-point channel codes with good error correction capabilities with *any* set of point-to-point channel codes with good shaping properties to construct coding schemes that can achieve the inner bounds for the different problems considered in this dissertation.

The following definitions will be used in the block-Markov constructions of coding schemes.

Definition 6.1.1 (Decoding Hamming Distance Spectrum)**.** Given a linear point-to-point channel code (H, ϕ) for a binary-input binary-output channel $p(y\,|\,x)$, the *decoding Hamming distance spectrum* W is defined as

$$W \triangleq d_H\big(V^n, \phi(V^n)\big),$$

where V^n is i.i.d. Bern(1/2), and $d_H(.,.)$ denotes the Hamming distance metric. Therefore, we have that $W \in \{0, 1, \dots, n\}$, and

$$\mathsf{P}\{W = w\} = \frac{\big|\{v^n \in \mathbb{F}_2^n : d_H\big(v^n, \phi(v^n)\big) = w\}\big|}{2^n},$$

for $0 \le w \le n$.

Definition 6.1.2 (Joint-Type)**.** Given a pair of sequences $(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n$, the *joint-type* of (x^n, y^n), denoted type(x^n, y^n), is the number of occurrence of each symbol pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ in (x^n, y^n), i.e.,

$$\text{type}(x^n, y^n) = (\pi(x, y\,|\,x^n, y^n))_{(x,y)\in\mathcal{X}\times\mathcal{Y}}\,,$$

where $\pi(x, y\,|\,x^n, y^n) = |\{i : 1 \le i \le n,\, x_i = x,\, y_i = y\}|$.

Definition 6.1.3 (Decoding Joint-Type Spectrum)**.** Consider a symmetric binary-input channel $p(y\,|\,x)$ and a linear point-to-point channel code (H, ϕ). Let Y^n be an i.i.d. according to $p(y)$, where $p(y) = \sum_x \frac{1}{2} p(y\,|\,x)$, and define $X^n = \phi(Y^n)$. The *decoding joint-type spectrum* T is defined as the joint-type of the pair (X^n, Y^n), i.e.,

$$T \triangleq \text{type}(X^n, Y^n).$$

For example, if $\mathcal{Y} = \{0, 1\}$, then $T = (T_{00}, T_{01}, T_{10}, T_{11})$, where T_{xy} is the number of occurrences of the symbol pair (x, y) in the sequences (X^n, Y^n), for $x, y \in \{0, 1\}$.

The main differences that can be observed in comparison with the previous constructions are the following.

- The modified constructions do not require any constituent codebooks to be nested, unlike the previous constructions which often assumed that the codebook corresponding to one Lego brick is a subcode of another.

- The modified coding schemes have a block-Markov structure, i.e., the inputs to one encoding/decoding block can depend on other coding blocks. Such a structure allows to share information about the transmitted sequences across coding blocks.
- The assumptions made on the Lego bricks of the modified constructions involve the distribution of either the decoding Hamming distance spectrum or the decoding joint-type spectrum of the constituent codes, both of which can be estimated efficiently in practice for any commercial off-the-shelf code. More specifically, the constructions here will imply that a decoding Hamming distance spectrum that is "close" to a Binomial distribution, or a decoding joint-type spectrum that is "close" to a Multinomial distribution are preferred, where "closeness" is measured via the total variation distance metric.
- A uniformly chosen random permutation that is shared between the encoder and the decoder is used in the modified constructions. The random permutation allows to shape the binary sequences according to the desired distribution, as will be shown in Lemma 6.2.1 and Lemma 6.3.1.

Although all the coding schemes that we considered have a block-Markov variant, for conciseness, we will focus on two problems: the asymmetric channel coding problem in Section 6.2 and the lossy source coding problem for an asymmetric source in Section 6.3. These two constructions already give the gist of the idea, and the extension to other coding problems becomes straightforward.

6.2 Asymmetric Channel Coding

Consider the problem of coding for an asymmetric channel $p(y\,|\,x)$, where the capacity-achieving input distribution is $p(x) \sim \mathrm{Bern}(\alpha)$ for some $\alpha \in (0, 1/2)$. The coding scheme we present here is defined upon the transmission of b blocks of information for some fixed b, and is constructed starting from the following three Lego bricks.

Lego Brick 6.2.1 (SW → Asym): *an $(n-k_1, n)$ linear Slepian–Wolf code (H_1, ϕ_1) for the problem $p(x, y) = p(x)p(y\,|\,x)$ with an average probability of error ϵ.*

Lego Brick 6.2.2 (P2P → Asym): *a (k_2, n) linear point-to-point channel code (H_2, ϕ_2) for* BSC(α) *with a decoding Hamming distance spectrum W_2 satisfying*

$$\frac{1}{2}\sum_{w=0}^{n}\left|\, P\{W_2 = w\} - \binom{n}{w}\alpha^w(1-\alpha)^{n-w}\right| \le \delta, \tag{6.1}$$

for some $\delta > 0$. We assume that $k_2 < k_1$.

Lego Brick 6.2.3 (P2P → Asym): *a (k_{sym}, n) point-to-point channel code $(f_{\mathrm{sym}}, \phi_{\mathrm{sym}})$ for the channel $p(y|x)$ with average probability of error ϵ_{sym}.*[1]

[1] For example, one can use any code that approaches the symmetric capacity of the channel. Note that this code will be used only in the first transmission block.

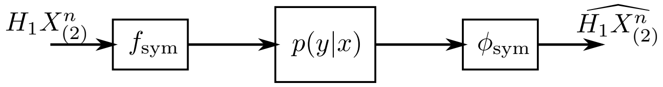


Figure 6.1: A block-Markov coding scheme for an asymmetric channel in the first transmission block using the code $(f_{\text{sym}}, \phi_{\text{sym}})$.

Remark 6.2.1. Since the alphabet size of W_2 is linear in n, the distribution of W_2 can be estimated for any off-the-shelf channel code via simulations. Hence, condition (6.1) is easily verifiable in practice. In words, this condition states that the distribution of the decoding Hamming distance spectrum is δ-away in total variation distance from a $\text{Binom}(n, \alpha)$ distribution.

Remark 6.2.2. Let $H_2 = \begin{bmatrix} A & B \end{bmatrix}$ for some nonsingular matrix B. Also, let $\widetilde{H}_2$ is as defined in (2.1).

Now, we are ready to describe the coding scheme. For each $j \in [b]$, let $V^n_{(j)}$ be an i.i.d. Bern(1/2) random dither shared between the encoder and the decoder, and let $\Gamma_{(j)} : [n] \to [n]$ be a permutation chosen uniformly at random and shared between the encoder and the decoder. We describe the encoding procedure starting from the bth block. Given a message $M_{(b)} \in \{0,1\}^{n-k_2}$, the encoder computes the sequence $Z^n_{(b)}$ as follows.

$$Z^n_{(b)} = \begin{bmatrix} \mathbf{0} \\ M_{(b)} \end{bmatrix}, \tag{6.2}$$

where $\mathbf{0}$ consists of k_2 zeros. Then, for each $j = b, \ldots, 2$, the encoder computes the sequences $\widetilde{X}^n_{(j)}$, $X^n_{(j)}$ and $Z^n_{(j-1)}$ as follows.

$$\begin{aligned} \widetilde{X}^n_{(j)} &= \phi_2\big(Z^n_{(j)} \oplus V^n_{(j)}\big) \oplus Z^n_{(j)} \oplus V^n_{(j)}, \\ X^n_{(j)} &= \Gamma_{(j)}\left(\widetilde{X}^n_{(j)}\right), \\ Z^n_{(j-1)} &= \begin{bmatrix} \mathbf{0} \\ H_1 X^n_{(j)} \\ M_{(j-1)} \end{bmatrix}, \end{aligned} \tag{6.3}$$

where $M_{(j-1)} \in \{0,1\}^{k_1-k_2}$. Note that the sequence $Z^n_{(j-1)}$ in the $(j-1)$-th transmission block depends on the syndrome vector corresponding to the channel input $X^n_{(j)}$ in the jth block. Furthermore, notice that the sequences $Z^n_{(j)}$ satisfy that $\widetilde{H}_2 Z^n_{(j)} = Z^n_{(j)}$ for each $j = 2, \ldots, b$. Finally, in the first block, the transmitter uses the encoder f_{sym} to encode the syndrome vector $H_1X^n_{(2)}$, where $X^n_{(2)}$ is the transmitted sequence in the second block. Fig. 6.1 and Fig. 6.2 show the block diagrams of the coding scheme in the first block and the remaining blocks, respectively. Note that a loss in the overall rate is incurred in the first block, where no message is transmitted. However, this loss decays as $1/b$, and, thus, by choosing b large enough, the rate loss becomes negligible.

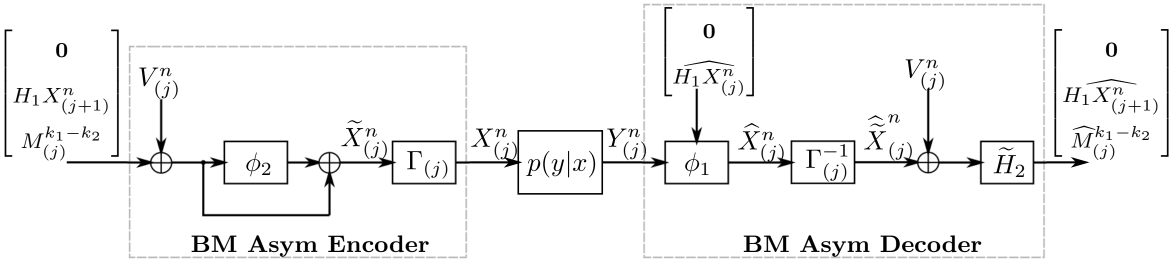


Figure 6.2: A block-Markov coding scheme for an asymmetric channel in the jth transmission block, for $2 \leq j \leq b-1$.

The following lemma states that under this construction, the distribution of the channel input $X^n_{(j)}$ is δ-away in total variation distance from the i.i.d. Bern(α) distribution, for each $j = 2, \ldots, b$.

Lemma 6.2.1. *For the arrangement shown in Fig. 6.2, we have*

$$\frac{1}{2}\sum_{x^n}\left|P\{X^n_{(j)} = x^n\} - \alpha^{\mathrm{wt}(x^n)}(1-\alpha)^{n-\mathrm{wt}(x^n)}\right| \leq \delta,$$

for each $j = 2, \ldots, b$.

Proof. See Appendix 6.A. □

At the decoder side, the coding scheme proceeds as follows. In the first transmission block, the decoder uses the decoder ϕ_{sym} to get an estimate $\widehat{H_1X^n_{(2)}}$ of $H_1X^n_{(2)}$. In the subsequent blocks, the decoder views $(X^n_{(j)}, Y^n_{(j)})$ as realizations of the Slepian–Wolf problem $p(x,y)$, uses the Slepian–Wolf decoder ϕ_1 to recover $X^n_{(j)}$ using the estimate $\widehat{H_1X^n_{(j)}}$ from previous blocks, and then reverse-engineers the operations at the encoder to get an estimate $\widehat{M_{(j)}}$ of the message in the jth block and an estimate $\widehat{H_1X^n_{(j+1)}}$ of the subsequent index.

For the probability of error of the coding scheme, consider the performance of a genie-aided decoder which recovers an estimate of $M_{(j)}$ in the jth block based on the channel output $Y^n_{(j)}$ and the syndrome vector $H_1X^n_{(j)}$ (which is supplied correctly by a genie regardless of any decoding errors in previous blocks). Notice that such a decoder would have the same probability of error over the b blocks as our decoder. To see this, observe that a decoding error can propagate from one block to another only through an error in the syndrome vector $H_1X^n_{(j)}$. Consider the first block where such an error happens. Both decoders would make an error in that block, which is precisely an error event over the b blocks, irrespective of decisions made in subsequent blocks. A similar argument has been made in the analysis of the successive cancellation decoding of polar codes [6]. Therefore, it suffices to analyze the error probability of the genie-aided decoder over the b blocks of transmission. Using similar bounding techniques as in Section 4.1.2, the average probability of error of the genie-aided decoder in the jth block can be bounded as

$$\mathsf{P}\{\widehat{M}_{(j)} \neq M_{(j)}\} \leq \delta + \epsilon,$$

and, thus, the average probability of error of our coding scheme over the b transmission blocks can be bounded using the union bound as

$$\mathsf{P}\left\{\widehat{M}_{(j)} \neq M_{(j)} \text{ for some } j \in [b]\right\} \leq (b-1)(\delta+\epsilon) + \epsilon_{\text{sym}},$$

where ϵ_{sym} is the average probability of error of the code used in the first block. The rate of the coding scheme is given by

$$R = \frac{(b-1)(k_1-k_2)+n-k_1}{nb}.$$

Remark 6.2.3. A point-to-point channel code with a decoding Hamming distance spectrum that is arbitrarily close in total variation distance to a $\text{Binom}(n, \alpha)$ distribution exists for sufficiently large n if and only if the rate k_2/n of the code is larger than $1 - H(\alpha) = 1 - H(X)$. To show this, the same argument of Section 2.3 based on distributed channel synthesis can be used. Similarly, a Slepian–Wolf code for $p(x, y)$ with arbitrarily small error probability exists for sufficiently large n if and only if its rate $\frac{n-k_1}{n}$ is larger than $H(X \,|\, Y)$. It follows that the rate R of the block-Markov asymmetric channel coding scheme can be made arbitrarily close to $I(X; Y)$ for sufficiently large n and b.

6.3 Lossy Source Coding

Now, we consider the problem of lossy source coding of a $\text{Bern}(\theta)$ source for some $\theta \in (0, 1/2)$, as defined in Section 3.2.1. In this part, we construct a coding scheme for this problem starting from a point-to-point channel code and a lossless source code that satisfy some easily-verifiable properties. In this case, an assumption about the *decoding joint-type spectrum* of the point-to-point channel code will be made. The coding scheme will be defined upon its operation on b blocks of source sequences. To this end, let D be some desired distortion level, and $p(x \,|\, \hat{x}) \sim \text{BSC}(D)$ be the desired conditional pmf of the source given the reconstruction. Under this conditional pmf, the distribution of the reconstruction is $p(\hat{x}) \sim \text{Bern}(\alpha)$, where

$$\alpha = \frac{\theta - D}{1 - 2D}.$$

The coding scheme is constructed using the following Lego bricks.

Lego Brick 6.3.1 (P2P → Lossy): *a (k_1, n) linear point-to-point channel code (H_1, ϕ_1) for the channel*

$$\bar{p}(x, v \,|\, \hat{x}) = p_{X,\widehat{X}}(x, \hat{x} \oplus v),$$

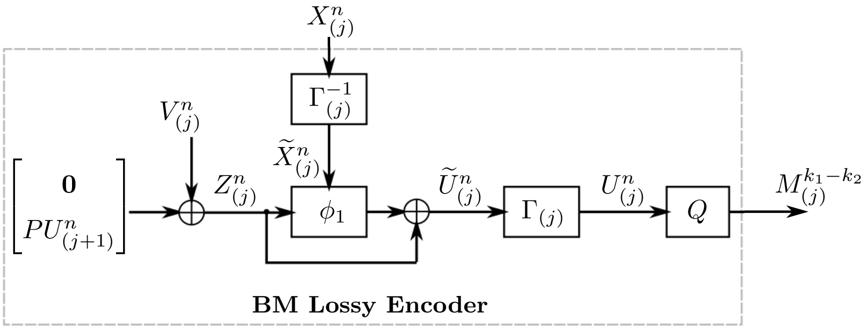


Figure 6.3: Encoder of the block-Markov lossy source coding scheme in the jth coding block, for $2 \leq j \leq b$.

such that the code's decoding joint-type spectrum T satisfies

$$d_{TV}\big(T, \text{Multinomial}(n; p_{000}, \ldots, p_{111})\big) \leq \delta \tag{6.4}$$

for some $\delta > 0$, where $d_{TV}(.,.)$ denotes the total variation distance and

$$p_{i_1 i_2 i_3} = \frac{1}{2} p_{X,\widehat{X}}(i_1, i_2 \oplus i_3), \qquad i_1, i_2, i_3 \in \{0, 1\}.$$

Lego Brick 6.3.2 (Lossless → Lossy): *an $(n - k_2, n)$ lossless source code (H_2, ϕ_2) for a* BSC(α) *source with average probability of error ϵ such that $k_2 < k_1$.*

Remark 6.3.1. Condition (6.4) says that the distribution of T is δ-away in total variation distance from a multinomial distribution. This condition can be easily verified in practice since the distribution of T can be estimated efficiently for any commercial off-the-shelf code. Recall that a multinomial random variable $M \sim \text{Multinomial}(n; p_1, \ldots, p_K)$ models the number of counts of each side of a K-sided die when rolled n times, where $p_1, \ldots, p_K$ are the probabilities of observing each side. We have

$$\mathsf{P}\{M = (m_1, \ldots, m_K)\} = \frac{n!}{m_1! \cdots m_K!} p_1^{m_1} \cdots p_K^{m_K}$$

for each $(m_1, \ldots, m_K)$ such that $\sum_{i=1}^{K} m_i = n$.

Remark 6.3.2. Let P and Q denote submatrices of H_2 such that $H_2 = \begin{bmatrix} P \\ Q \end{bmatrix}$, for some $(n - k_1) \times n$ matrix P and $(k_1 - k_2) \times n$ matrix Q.

Fig. 6.3 shows the encoding scheme of the lossy source code, where $V^n_{(j)}$ is a random dither shared with the decoder, and $\Gamma_{(j)}$ is a random permutation chosen uniformly at random and shared with the decoder. Similar to the coding scheme of Section 3.2.3, the main idea is to use the available Lego bricks to generate a sequence $U^n_{(j)}$ with a distribution that is "close" to the i.i.d. $p(\hat{x})$ distribution and then convey the sequence "losslessly" to the decoder.

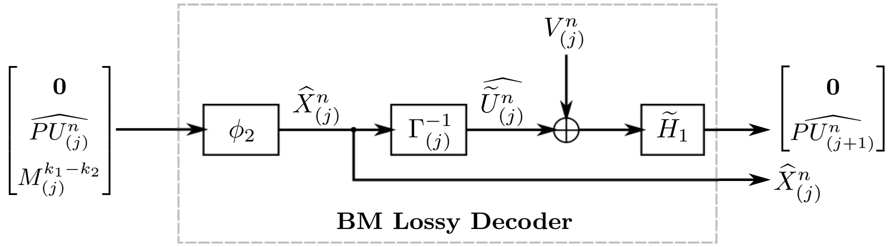


Figure 6.4: Decoder of the block-Markov lossy source code in the jth coding block, for $2 \leq j \leq b-1$.

The encoding procedure starts from the bth block. For each $j = b, \ldots, 2$, the encoder computes the sequences $Z^n_{(j)}$, $\widetilde{U}^n_{(j)}$, and $U^n_{(j)}$ as follows.

$$\begin{aligned} Z^n_{(j)} &= \begin{bmatrix} \mathbf{0} \\ PU^n_{(j+1)} \end{bmatrix} \oplus V^n_{(j)}, \\ \widetilde{U}^n_{(j)} &= \phi_1\left(\Gamma^{-1}_{(j)}(X^n_{(j)}), Z^n_{(j)}\right) \oplus Z^n_{(j)}, \\ U^n_{(j)} &= \Gamma_{(j)}(\widetilde{U}^n_{(j)}), \end{aligned} \tag{6.5}$$

where $PU^n_{(b+1)} = \mathbf{0}^{n-k_1}$. Notice that, for each $j = 2, \ldots, b$, we have

$$\widetilde{H}_1\widetilde{U}^n_{(j)} = \widetilde{H}_1 Z^n_{(j)} = \begin{bmatrix} \mathbf{0} \\ PU^n_{(j+1)} \end{bmatrix} \oplus \widetilde{H}_1 V^n_{(j)}. \tag{6.6}$$

The next lemma states that the distribution of $(X^n_{(j)}, U^n_{(j)})$ in Fig. 6.3 is δ-away in total variation distance from the i.i.d. $p(x, \hat{x})$ distribution.

Lemma 6.3.1. *Let $q(x^n, u^n)$ be the true distribution of $(X^n_{(j)}, U^n_{(j)})$. Then, we have*

$$\frac{1}{2}\sum_{x^n,u^n}\left|q(x^n,u^n) - \prod_{i=1}^n p_{X,\widehat{X}}(x_i,u_i)\right| \leq \delta,$$

for each $j = 2, \ldots, b$.

Proof. See Appendix 6.B. □

Fig. 6.4 shows the decoding scheme of the lossy source code. The decoder uses the lossless source decoder and the estimate $\widehat{PU^n_{(j)}}$ of $PU^n_{(j)}$ which the decoder had found in previous coding blocks to compute the reconstruction sequence $\widehat{X}^n_{(j)}$. The decoder then reverse-engineers the operations at the encoder side and computes an estimate $\widehat{PU^n_{(j+1)}}$ for the subsequent coding block using the observation of (6.6), as shown in Fig. 6.4. Using similar arguments as in Section 3.2.3, it can be shown that the average distortion of the coding scheme over the b blocks can be bounded as

$$\frac{1}{b-1}\sum_{j=2}^{b}\frac{1}{n}\,\mathsf{E}[d_H(X^n_{(j)}, \widehat{X}^n_{(j)})] \leq D + \delta + b\epsilon,$$

where the factor b in front of ϵ pertains to the use of the union bound on the error probability over b blocks. In this analysis, we assume that in the first coding block, $PU^n_{(2)}$ is passed to the decoder error-free. The rate of the coding scheme is

$$R = \frac{(b-1)(k_1 - k_2) + n - k_1}{(b-1)n}.$$

Remark 6.3.3. A point-to-point channel code with a decoding joint-type spectrum that is arbitrarily close in total variation distance to a multinomial distribution exists for sufficiently large n if and only if the rate k_1/n of the code is larger than $1 - H(\widehat{X}\,|\,X)$. A similar argument as in Section 2.3 based on distributed channel synthesis can be used to show this. Similarly, a lossless source code for a Bern(α) source with arbitrarily small error probability exists for sufficiently large n if and only if its rate $\frac{n-k_2}{n}$ is larger than $H(\alpha) = H(\widehat{X})$. It follows that the rate R of the block-Markov lossy source coding scheme can be made arbitrarily close to $I(X;\widehat{X})$ for sufficiently large n and b.

6.4 Other Coding Problems

The two coding schemes presented in Sections 6.2 and 6.3 for the asymmetric channel coding problem and the lossy source coding problem were constructed starting from Lego bricks that satisfy some easily-verifiable properties. In a similar way, all other coding schemes that we have seen in Chapters 3, 4 and 5 can be modified to start from such Lego bricks. For all coding schemes, the constructions will have a window size of 2, i.e., the encoding and decoding of the current sequence block will depend only on either the previous block or the subsequent block. This is favorable from a practical point of view as it allows to reduce the space complexity of the algorithms.

Since the block-Markov constructions do not require any constituent codebooks to be nested, it follows that one can assemble *any* set of point-to-point channel codes with good error correction capabilities (i.e., small probability of error) with *any* set of point-to-point channel codes with good shaping properties (i.e., a decoding Hamming distance spectrum that is “close” to a Binomial, or a decoding joint-type spectrum that is “close” to a Multinomial) to achieve the best known inner bounds for the problems discussed in this dissertation.

6.5 Concluding Remarks on the Lego-Brick Approach

The Lego-brick framework we developed in Chapters 2–6 is promising, and provides guidelines to implement practical coding schemes for network communication. The proposed code constructions can achieve larger rates in theory and demonstrate better performance in practice compared to the naive approach of coding over the point-to-point links of a network. Moreover, the computational complexity of the proposed constructions is governed by the encoding/decoding complexities of the constituent Lego bricks, which can be taken “off-the-shelf”; hence, the constructions are friendly to hardware implementation using existing coding blocks.

Two interesting observations that can be made regarding the proposed constructions are the following. First, one can see some form of duality between some of the constructions. For example, the encoder of the asymmetric channel coding scheme looks almost identical to the decoder of the lossy source coding scheme, and vice versa. Such a relation is well-known in the literature and has been pointed out in [49] and [104]. Nonetheless, note that, in our construction, the asymmetric channel encoder has a "shaping role", whereas the lossy source decoder has an "error correction role". Although not mathematically precise, this suggests some form of duality between error correction and shaping in our constructions, which can be seen as analogous to the well-understood packing-and-covering duality in network information theory. Similar observations can be made between the Gelfand–Pinkser coding scheme and the Wyner–Ziv coding scheme, between the Marton coding scheme and the Berger–Tung coding scheme, as well as between the downlink and uplink C-RAN code constructions. Secondly, as is apparent from most of our constructions, randomization plays an important role in the design of coding schemes for network communication. The Lego bricks were often used alongside a "dithering" brick, which generates a binary sequence uniformly at random, or an "interleaver" brick, which applies a permutation chosen uniformly at random to a binary sequence. Such randomness can shape signals into a desired structure, and will often be assumed to be shared between the encoders and decoders. Note that sharing randomness between the transmitter and the receiver is a common practice in wireless communications, e.g., sharing the (pseudo)random spreading code in code-division multiple access (CDMA) systems [105] or sharing the interleaving pattern in 5G NR systems [29].

We end this part of the dissertation by discussing some ideas for future work.

- First, most of our constructions required that the channels are binary-input and/or the sources are binary sources. It is of particular interest to extend these constructions to channels with **non-binary** channel inputs and sources that are non-binary; this would allow one to establish a general framework for implementing **coded modulation** techniques for multiuser settings.

- Secondly, all the constructions that we presented in this dissertation allow to transmit at rates that were already known to be information-theoretically achievable. In doing so, we utilized tools such as nested linear structures, linear codes with good shaping properties, and block-Markov coding. The next challenge is to exploit these practically-motivated building blocks in order to achieve **new and tighter inner bounds** for open problems in network information theory.

- Thirdly, to the best of the knowledge of the author, apart from polar codes, there are no other known low-complexity point-to-point channel codes with a provably vanishing decoding distance. It would be interesting to **study how other codes perform in the decoding distance regime** (compared to the more-prominently studied error probability regime). Ultimately, the goal would be to **construct new families of point-to-point channel codes** with the property of a **vanishing decoding distance**. As we saw, this property can be crucial for constructing rate-optimal coding schemes in multiuser settings.

With these ideas, we conclude our study on the Lego-brick approach to coding for network communication. In the next chapter, we develop channel coding techniques for a different communication setting, namely, point-to-point channels with memory.

Acknowledgement

Chapter 6, in part, is a reprint with permission of the material as it appears in the papers: Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to coding for network communication,” arXiv:2211.07208, November 2022, which is in preparation to be submitted to *IEEE Transactions on Information Theory*, Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to lossy source coding,” in *2022 17th Canadian Workshop on Information Theory (CWIT)*, pp. 45–50, 2022, and Nadim Ghaddar, Shouvik Ganguly, Lele Wang, and Young-Han Kim, “A Lego-brick approach to coding for asymmetric channels and channels with state,” in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 1367–1372, 2021. The dissertation author was the primary investigator and author of these papers. This work was supported in part by the Institute for Information & Communication Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-01659, 5G Open Intelligence-Defined RAN (ID-RAN) Technique based on 5G New Radio), in part by the NSERC Discovery Grant No. RGPIN-2019-05448, and in part by the NSERC Collaborative Research and Development Grant CRDPJ 543676-19.

Appendix

6.A Proof of Lemma 6.2.1

Consider the block diagram shown in Fig. 6.2. Since $Z^n_{(j)} \oplus V^n_{(j)}$ is i.i.d Bern(1/2) for each $j = 2, \ldots, b$, then the sequence $\widetilde{X}^n_{(j)}$ satisfies that

$$\mathrm{wt}(\widetilde{X}^n_{(j)}) = d_H\big(\phi_2(Z^n_{(j)} \oplus V^n_{(j)}), Z^n_{(j)} \oplus V^n_{(j)}\big) \overset{d}{=} W_2,$$

where $\overset{d}{=}$ denotes equality in distribution and W_2 is the decoding Hamming distance spectrum of the code (H_2, ϕ_2). It follows that, for any sequence x^n with $\mathrm{wt}(x^n) = w$, we have that

$$\begin{aligned}
\mathsf{P}\{X^n_{(j)} = x^n\} &= \sum_{\widetilde{x}^n : \mathrm{wt}(\widetilde{x}^n) = w} \mathsf{P}\{\Gamma_{(j)}(\widetilde{x}^n) = x^n\}\, \mathsf{P}\{\widetilde{X}^n_{(j)} = \widetilde{x}^n\} \\
&= \sum_{\widetilde{x}^n : \mathrm{wt}(\widetilde{x}^n) = w} \frac{w!(n-w)!}{n!}\, \mathsf{P}\{\widetilde{X}^n_{(j)} = \widetilde{x}^n\} \\
&= \frac{1}{\binom{n}{w}}\, \mathsf{P}\{\mathrm{wt}\big(\widetilde{X}^n_{(j)}\big) = w\} \\
&= \frac{1}{\binom{n}{w}}\, \mathsf{P}\{W_2 = w\}.
\end{aligned}$$

Hence,

$$\begin{aligned}
\frac{1}{2}\sum_{x^n} \Big| \mathsf{P}\{X^n_{(j)} = x^n\} - \alpha^{\mathrm{wt}(x^n)}(1-\alpha)^{n-\mathrm{wt}(x^n)} \Big| &= \frac{1}{2}\sum_{x^n} \left| \frac{\mathsf{P}\{W_2 = \mathrm{wt}(x^n)\}}{\binom{n}{\mathrm{wt}(x^n)}} - \alpha^{\mathrm{wt}(x^n)}\big(1-\alpha\big)^{n-\mathrm{wt}(x^n)} \right| \\
&= \frac{1}{2}\sum_{w=0}^{n} \left| \mathsf{P}\{W_2 = w\} - \binom{n}{w}\alpha^w(1-\alpha)^{n-w} \right| \\
&\le \delta,
\end{aligned}$$

where the last step holds since W_2 satisfies condition (6.1).

6.B Proof of Lemma 6.3.1

Consider the encoding scheme shown in Fig. 6.3. Denote $C^n_{(j)} = \phi_1(\widetilde{X}^n_{(j)}, Z^n_{(j)})$. Since $\widetilde{X}^n_{(j)}$ is i.i.d. according to $p(x)$ and $Z^n_{(j)}$ is i.i.d. Bern(1/2), and $(\widetilde{X}^n_{(j)}, Z^n_{(j)})$ are independent, we have that

$$\text{type}\left(C^n_{(j)}, (\widetilde{X}^n_{(j)}, Z^n_{(j)})\right)$$

satisfies condition (6.4). Since $\widetilde{U}^n_{(j)} = C^n_{(j)} \oplus Z^n_{(j)}$, it follows using standard bounding techniques that

$$d_{TV}\left(\text{type}(\widetilde{X}^n_{(j)}, \widetilde{U}^n_{(j)}),\ \text{Multinomial}(n; p_{00}, p_{01}, p_{10}, p_{11})\right) \le \delta, \tag{6.7}$$

where $p_{ij} = p_{X,\widehat{X}}(i,j)$, for $i,j \in \{0,1\}$.

Now, let (x^n, u^n) be a pair of sequences such that $\text{type}(x^n, u^n) = t$, where $t \triangleq (t_{00}, t_{01}, t_{10}, t_{11})$ such that $\sum_{i,j} t_{ij} = n$. Then, we have that

$$\begin{aligned} q(x^n, u^n) &= \sum_{\substack{(\widetilde{x}^n, \widetilde{u}^n): \\ \text{type}(\widetilde{x}^n, \widetilde{u}^n) = t}} \mathsf{P}\{\Gamma(\widetilde{x}^n) = x^n, \Gamma(\widetilde{u}^n) = u^n\}\, \mathsf{P}\{\widetilde{X}^n_{(j)} = \widetilde{x}^n, \widetilde{U}^n_{(j)} = \widetilde{u}^n\} \\ &= \sum_{\substack{(\widetilde{x}^n, \widetilde{u}^n): \\ \text{type}(\widetilde{x}^n, \widetilde{u}^n) = t}} \frac{t_{00}!t_{01}!t_{10}!t_{11}!}{n!}\, \mathsf{P}\{\widetilde{X}^n_{(j)} = \widetilde{x}^n, \widetilde{U}^n_{(j)} = \widetilde{u}^n\} \\ &= \frac{t_{00}!t_{01}!t_{10}!t_{11}!}{n!}\, \mathsf{P}\left\{\text{type}(\widetilde{X}^n_{(t)}, \widetilde{U}^n_{(t)}) = t\right\}. \end{aligned}$$

Therefore,

$$\begin{aligned} &\frac{1}{2}\sum_{x^n, u^n} \left| q(x^n, u^n) - \prod_{i=1}^n p_{X,\widehat{X}}(x_i, u_i) \right| = \frac{1}{2} \sum_{\substack{t=(t_{00},t_{01},t_{10},t_{11}): \\ \sum t_{ij} = n}} \sum_{\substack{x^n, u^n: \\ \text{type}(x^n, u^n) = t}} \left| q(x^n, u^n) - \prod_{i=1}^n p_{X,\widehat{X}}(x_i, u_i) \right| \\ &= \frac{1}{2} \sum_{\substack{t=(t_{00},t_{01},t_{10},t_{11}): \\ \sum t_{ij} = n}} \sum_{\substack{x^n, u^n: \\ \text{type}(x^n, u^n) = t}} \left| \frac{t_{00}!t_{01}!t_{10}!t_{11}!}{n!}\, \mathsf{P}\left\{\text{type}(\widetilde{X}^n_{(t)}, \widetilde{U}^n_{(t)}) = t\right\} - \prod_{i,j=0}^{1} p_{ij}^{t_{ij}} \right| \\ &= \frac{1}{2} \sum_{\substack{t=(t_{00},t_{01},t_{10},t_{11}): \\ \sum t_{ij} = n}} \left| \mathsf{P}\left\{\text{type}(\widetilde{X}^n_{(t)}, \widetilde{U}^n_{(t)}) = t\right\} - \frac{n!}{t_{00}!t_{01}!t_{10}!t_{11}!} \prod_{i,j=0}^{1} p_{ij}^{t_{ij}} \right| \\ &\le \delta, \end{aligned}$$

where the last inequality holds from (6.7).

Chapter 7

Joint Channel Estimation and Polar Coding over Channels with Memory

In this chapter, we shift our attention to channels with memory. We devise a polar-coding-based strategy for joint channel estimation and coding over several channel models with memory. We show that the exploitation of the knowledge of the channel state distribution in channel decoding can result in substantial decoding performance gains. In Section 7.1, we describe the different channel models considered in this work, namely, the finite-state Markov channel, the Gauss-Markov channel, and the flat-fading channel. In Section 7.2, we briefly review some basics of polar coding and set up the necessary notation. In Section 7.3, the SCTD algorithm will be described for finite-state Markov channels of any finite order. The algorithm is given in sufficient detail that can allow for an independent implementation. Section 7.4 presents a joint estimation and decoding scheme that consists of an "estimation-aware" decoding algorithm for the three channel models, and an iterative procedure of estimation and decoding that incorporates reliable bits into subsequent iterations of channel estimation. On the other hand, Section 7.5 describes a joint piloting and encoding scheme consisting of a pilot arrangement pattern based on shortened polar codes. Section 7.6 is devoted to our simulation experiments over the different channel models. Finally, Section 7.7 concludes our work on channels with memory. Let's start with the channel models.

7.1 Channel Models

In this section, the three channel models considered in this chapter are defined. The channels in all three models are binary-input with channel input alphabet $\mathcal{X} = \{0, 1\}$, channel state alphabet $\mathcal{H}$ and channel output alphabet $\mathcal{Y}$. $\mathcal{H}$ is finite in the finite-state channel model (Section 7.1.1), whereas $\mathcal{H} = \mathbb{R}$ in the Gauss-Markov channel model (Section 7.1.2), and $\mathcal{H} = \mathbb{C}$ in the fading channel model (Section 7.1.3), where $\mathbb{R}$ and $\mathbb{C}$ are the sets of real and complex numbers, respectively. We consider N uses of the channels. With a slight abuse of notation, we denote by X_1^N, Y_1^N and H_1^N the channel input, channel output and channel state vectors, respectively, in all three cases. Throughout this chapter, we will assume that X_1^N and H_1^N are mutually independent, i.e., the transmitter has no channel state information.

7.1.1 Finite-State Markov Channel

The first channel model that is considered is a finite-state Mth-order Markov channel model W, where the channel state alphabet $\mathcal{H}$ is finite, and the channel state process $(H_n)_{n\geq 0}$ is an Mth order Markov chain. We denote the initial M states of the Markov chain by the M-dimensional random vector $H_i^M = (H_{1-M}, \dots, H_{-1}, H_0)$. The distribution of H_i^M will be denoted by π. Recall that Mth order Markovity implies that for each $n > 0$ and any sequence $h_{1-M}^n \in \mathcal{H}^{n+M}$, we have that

$$\mathsf{P}\{H_n = h_n | H_{1-M}^{n-1} = h_{1-M}^{n-1}\} = \mathsf{P}\{H_n = h_n | H_{n-M}^{n-1} = h_{n-M}^{n-1}\}.$$

The probability of observing $y \in \mathcal{Y}$, given that $x \in \mathcal{X}$ is transmitted and $h \in \mathcal{H}$ is the channel state, will be denoted by $W(y|x,h)$. Note that the channel output alphabet $\mathcal{Y}$ is not necessarily finite. Finite-state Markov channels have been widely used to model error bursts in wireline telephone networks [106, 107] and later on to model fading channels in mobile radio communications [108].

7.1.2 Gauss-Markov Channel

In a Gauss-Markov channel model, the channel output at time i in a block of length N can be expressed as

$$Y_i = H_i \mathrm{s}(X_i) + Z_i, \quad i = 1, \dots, N,$$

where $X_i \in \mathcal{X}$, $\mathrm{s} : \{0,1\} \to \{1,-1\}$ is the mapping corresponding to antipodal modulation, $Z_i \in \mathbb{R}$ follows a discrete-time Gaussian noise process with covariance matrix $\sigma_z^2 I$, and $H_i \in \mathbb{R}$ is the channel state and is modeled as a first-order Gauss-Markov process with parameter $\eta \in (0,1)$, i.e.,

$$H_i = \eta H_{i-1} + W_i, \quad i = 1, \dots, N,$$

where $W_1, W_2, \dots$ are i.i.d, zero-mean Gaussians with variance σ_w^2. Notice that the channel state here can take any value on the real line, and thus, is not restricted to a finite set. Such a model has been previously considered in [109, 110] to model the temporal correlation in fading channels.

7.1.3 Fading Channel

We consider Clarke's isotropic scattering model of the flat-fading channel [111], which is one of the most widely accepted statistical models of the fading random process. In this model, the channel state H_i is complex-valued, with independent in-phase and quadrature components I_i and Q_i, each following a Gaussian process with mean $\frac{\mu}{\sqrt{2}}$ and autocovariance function given by

$$R_I[k] = R_Q[k] = \sigma_h^2 J_0(2\pi f_m k), \tag{7.1}$$

where σ_h^2 is the variance of the Gaussian processes, $f_m = f_D T_s$ is the maximum Doppler frequency normalized by the sampling rate $1/T_s$, and $J_0(.)$ is the zero-th order Bessel function of the first kind. Note that the envelope of this complex-valued Gaussian process is a stationary process whose first-order distribution is Rician with shape parameter $\rho = \frac{\mu^2}{2\sigma_h^2}$ and scale parameter $\Omega = \mu^2 + 2\sigma_h^2$.[1] When $\mu = 0$, the model corresponds to the Rayleigh flat-fading channel [112, Chapter 9]. Given that the channel state H_i follows this model, the channel output at time index i in a block of length N can be written as

$$Y_i = H_i \mathrm{s}(X_i) + Z_i, \quad i = 1, \ldots, N,$$

where $X_i \in \mathcal{X}$, $\mathrm{s} : \{0, 1\} \to \{1, -1\}$ is the mapping corresponding to antipodal modulation, and Z_i is a zero-mean, complex-valued, circularly-symmetric Gaussian random variable with variance σ_z^2.

7.2 Polar Coding Preliminaries

Consider a polar code of block length $N = 2^n$ and information set $\mathcal{A}$, with $|\mathcal{A}| = K$, i.e., the number of frozen bits is $N - K$. The code rate is therefore $R_{code} = \frac{K}{N}$. We denote by U_1^N the information bit vector (which includes frozen bits), and by X_1^N the corresponding codeword, which is transmitted over a binary-input channel.

The matrix G_n is used to denote the n-fold Kronecker product of the matrix $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. G_n is referred to as the *base matrix* for a polar code of length $N = 2^n$. Note that the generator matrix of the polar code is the submatrix of G_n formed by taking the rows with indices belonging to $\mathcal{A}$. One basic property of G_n is that all base matrices of polar codes with block lengths less than

[1] The shape parameter can be understood as the ratio between the power in the direct line-of-sight path and the power in other scattered paths. The scale parameter is understood as the total power in all paths.

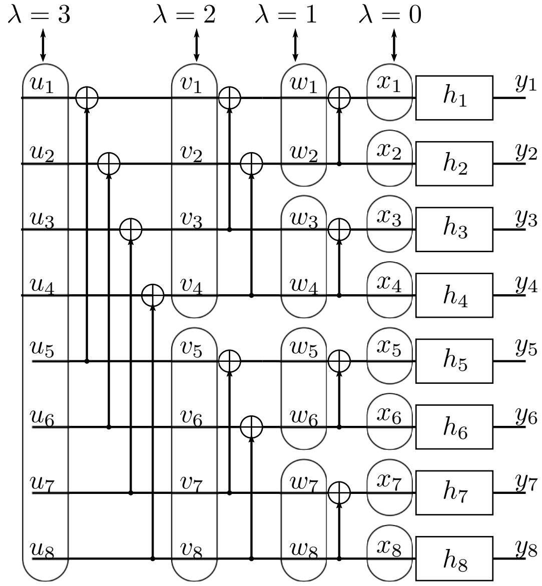


Figure 7.1: Polar transformation for a code of length $N = 8$.

N, i.e., $G_{n-i}, i = 1, \dots, n-1$, appear as sub-matrices of G_n, e.g.

$$\begin{aligned} G_n &= \begin{bmatrix} G_{n-1} & \mathbf{0} \\ G_{n-1} & G_{n-1} \end{bmatrix} \\ &= \begin{bmatrix} G_{n-2} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ G_{n-2} & G_{n-2} & \mathbf{0} & \mathbf{0} \\ G_{n-2} & \mathbf{0} & G_{n-2} & \mathbf{0} \\ G_{n-2} & G_{n-2} & G_{n-2} & G_{n-2} \end{bmatrix} = \cdots , \end{aligned} \tag{7.2}$$

where $\mathbf{0}$ denotes the all-zero matrix of the appropriate dimension. This simple property will turn out to be useful in the description of the pilot arrangement pattern based on shortened polar codes in Section 7.5. Figure 7.1 shows the polar transformation based on G_3 for a code of length $N = 8$.

We proceed now to define some notation that will be useful in our later description of decoding algorithms. First, we associate each bit in the polar transformation to a variable B, indexed by a triplet (λ, ϕ, β), where

$$\begin{aligned} &0 \le \lambda \le n, \\ &1 \le \phi \le 2^{\lambda}, \\ &1 \le \beta \le 2^{n-\lambda}. \end{aligned} \tag{7.3}$$

Thus, the bit variable will be denoted by $B_{\lambda,\phi,\beta}$. That is, for any triplet (λ, ϕ, β) that satisfies (7.3), $B_{\lambda,\phi,\beta}$ corresponds to a particular bit in the polar transformation. For example, in Figure 7.1, $u_1^8 = (B_{3,\phi,1})_{\phi=1}^8$ and $x_1^8 = (B_{0,1,\beta})_{\beta=1}^8$, and bits in different circled regions at a particular λ correspond to different values of β. In order to aid in the exposition, we will refer hereafter to index

λ as the *layer*, index ϕ as the *phase* and index β as the *branch*[2]. Also, for the sake of brevity, we will often use the shorthand notation

$$\Lambda = 2^{\lambda}.$$

In addition to the bit variable $B_{\lambda,\phi,\beta}$, we define a probability array variable P, indexed by a triplet (λ, ϕ, β) satisfying (7.3) and a bit $b \in \{0, 1\}$, which will be denoted by $P_{\lambda,\phi,\beta}[b]$. Analogously, for Arıkan's successive cancellation decoder over memoryless channels, one can think of $P_{\lambda,\phi,\beta}[b]$ as the likelihood probabilities of the synthetic bit-channels [6, Equations (17) and (18)]. Here, nonetheless, $P_{\lambda,\phi,\beta}[b]$ will correspond to an *array* of probability values that are associated with the event that the bit $B_{\lambda,\phi,\beta}$ takes value b. The array size of $P_{\lambda,\phi,\beta}[b]$ and the analytical expression it corresponds to will vary in different sections of this chapter. In each case, $P_{\lambda,\phi,\beta}[b]$ will be explicitly defined.

7.3 Successive Cancellation Trellis Decoding

After Arıkan's seminal work on polar coding for BMS channels, attempts followed to apply polar codes to channels with memory. In [114, 115], Şaşoğlu showed that Arıkan's recursive construction polarizes a large class of processes with memory, including ones corresponding to finite-state Markov channels of any arbitrary finite order. More recently, the authors in [116] showed that the polarization of Markov channels happens at the same rate as in the memoryless setting. This result implies that polar codes can attain vanishing error probability over these channels. In [117, 118], a practical polar decoding algorithm for finite-state first-order Markov channels was proposed, namely, the successive cancellation trellis decoder (SCTD). The algorithm generalizes successive cancellation decoding, taking into account the Markov property of the channel state. In this section, we provide a detailed description of the SCTD algorithm. In comparison to [117] and [118], our description extends the algorithm to finite-state Markov channels of any finite memory order.

To this end, we will assume that H_1^N follows a finite-state Mth order Markov chain (Section 7.1.1). The distribution π of the initial M states of the Markov chain is assumed to be known to the decoder. As in the memoryless setting, SCTD decoding defines synthetic channels $W_\lambda^{(\phi,\beta)}$, the recursive structure of which assists in an efficient implementation of the decoder. More precisely, for a triplet (λ, ϕ, β) satisfying (7.3), the channel $W_\lambda^{(\phi,\beta)}$ has input alphabet $\mathcal{X} \times \mathcal{H}^M$ and output alphabet $\mathcal{Y}^\Lambda \times \mathcal{X}^{\phi-1} \times \mathcal{H}^M$, and its conditional probability will be generally denoted by[3]

$$W_\lambda^{(\phi,\beta)}(y^\Lambda, u_1^{\phi-1}, h_f^M | u_\phi, h_i^M),$$

where $h_i^M \triangleq h_{(\beta-1)\Lambda-M+1}^{(\beta-1)\Lambda}$ and $h_f^M \triangleq h_{\beta\Lambda-M+1}^{\beta\Lambda}$ represent the initial and final M states corresponding to layer λ and branch β, where the dependence on (λ, β) is dropped for notation convenience. Also,

[2] We follow the same nomenclature as Tal and Vardy's list decoding paper [113].

[3] In our description, we will assume that $\mathcal{Y}$ is a countable set, yet the decoder for the case when $\mathcal{Y}$ is uncountable follows by simply replacing conditional probabilities with the corresponding densities.

although not particularly shown in the notation, y^Λ, here, represents a contiguous subvector of the channel output corresponding to layer λ and branch β, i.e., $y^\Lambda \triangleq y_{(\beta-1)\Lambda+1}^{\beta\Lambda}$. As an example, in Figure 7.1, if $M = 2$, then at layer $\lambda = 2$ and branch $\beta = 2$, we would have $y^\Lambda = y_5^8$, $h_i^M = (h_3, h_4)$, and $h_f^M = (h_7, h_8)$.

Now, we can proceed to describe the decoding algorithm. Recall that the fundamental requirement in any polar decoder is to compute, for each $1 \leq \phi \leq N$, the probabilities $\mathsf{P}\{Y_1^N = y_1^N, U_1^{\phi-1} = \hat{u}_1^{\phi-1} \,|\, U_\phi = u_\phi\}$, which can be expressed by

$$
\begin{aligned}
&\mathsf{P}\{Y_1^N = y_1^N, U_1^{\phi-1} = \hat{u}_1^{\phi-1} \,|\, U_\phi = u_\phi\} \\
&= \sum_{\substack{(h_i^M, h_f^M) \\ \in \mathcal{H}^{2M}}} \pi(h_i^M)\, \mathsf{P}\{Y_1^N = y_1^N, U_1^{\phi-1} = \hat{u}_1^{\phi-1}, H_{N-M+1}^N = h_f^M | U_\phi = u_\phi, H_i^M = h_i^M\} \\
&= \sum_{\substack{(h_i^M, h_f^M) \\ \in \mathcal{H}^{2M}}} \pi(h_i^M) W_n^{(\phi,1)}(y_1^N, \hat{u}_1^{\phi-1}, h_f^M | u_\phi, h_i^M)
\end{aligned}
\tag{7.4}
$$

The main idea in SCTD is that the conditional probabilities $W_\lambda^{(\phi,\beta)}(y^\Lambda, u_1^{\phi-1}, h_f^M | u_\phi, h_i^M)$ can be recursively computed, starting from the "base condition"

$$
W_0^{(1,\beta)}(y_\beta, h_\beta \,|\, x_\beta, h_{\beta-M}^{\beta-1}) = p(h_\beta \,|\, h_{\beta-M}^{\beta-1}) W(y_\beta \,|\, x_\beta, h_\beta), \tag{7.5}
$$

where $1 \leq \beta \leq N$ and $p(h_\beta | h_{\beta-M}^{\beta-1})$ are the transition probabilities of the Mth-order Markov chain. The recursion is formalized in Theorem 1, which can be seen as a recast of [119, Theorem 2] for Markov channels of any order.

Theorem 1. *Let (λ, ϕ, β) satisfy (7.3), and let $\psi = \lceil \phi/2 \rceil$. Then*

$$
\begin{aligned}
&W_\lambda^{(2\psi-1,\beta)}(y_1^\Lambda, u_1^{2\psi-2}, h_f^M \,|\, u_{2\psi-1}, h_i^M) \\
&= \sum_{\tilde{h}^M \in \mathcal{H}^M} \left(\sum_{u_{2\psi}} \frac{1}{2} \Big(W_{\lambda-1}^{(\psi,2\beta-1)}(y_1^{\Lambda/2}, u_{1,o}^{2\psi-2} \oplus u_{1,e}^{2\psi-2}, \tilde{h}^M \,|\, u_{2\psi-1} \oplus u_{2\psi}, h_i^M) \right. \\
&\qquad\qquad \left. W_{\lambda-1}^{(\psi,2\beta)}(y_{\Lambda/2+1}^\Lambda, u_{1,e}^{2\psi-2}, h_f^M \,|\, u_{2\psi}, \tilde{h}^M) \Big) \right),
\end{aligned}
\tag{7.6}
$$

$$
\begin{aligned}
&W_\lambda^{(2\psi,\beta)}(y_1^\Lambda, u_1^{2\psi-1}, h_f^M \,|\, u_{2\psi}, h_i^M) \\
&= \sum_{\tilde{h}^M \in \mathcal{H}^M} \frac{1}{2} \Big(W_{\lambda-1}^{(\psi,2\beta-1)}(y_1^{\Lambda/2}, u_{1,o}^{2\psi-2} \oplus u_{1,e}^{2\psi-2}, \tilde{h}^M \,|\, u_{2\psi-1} \oplus u_{2\psi}, h_i^M) \\
&\qquad\qquad W_{\lambda-1}^{(\psi,2\beta)}(y_{\Lambda/2+1}^\Lambda, u_{1,e}^{2\psi-2}, h_f^M \,|\, u_{2\psi}, \tilde{h}^M) \Big),
\end{aligned}
\tag{7.7}
$$

where $u_{1,o}^{2\psi}$ and $u_{1,e}^{2\psi}$ denote subsequences of $u_1^{2\psi}$ consisting of odd and even indices, respectively.

Proof. The proof follows in the same way as [118, Theorem 2], while noting that for an Mth-order Markov channel, $(Y_{\Lambda/2+1}^\Lambda, X_{\Lambda/2+1}^\Lambda, H_{\Lambda-M+1}^\Lambda)$ is conditionally independent of $(Y_1^{\Lambda/2}, X_1^{\Lambda/2}, H_i^M)$, given $H_{\Lambda/2-M+1}^{\Lambda/2}$, for any $0 \leq \lambda \leq n$ and $\Lambda = 2^\lambda$. □

Algorithm 1: Successive Cancellation Trellis Decoder

```
Input: Received vector $y_1^N$, information set $\mathcal{A}$.
Output: Decoded information vector $\hat{u}^N$, Decoded codeword $\hat{x}^N$, Bit-channel
    likelihoods $L_{out}$.
for $\beta = 1, \dots, N$ do                                    // base condition (7.5)
    for $h_i^M \in \mathcal{H}^M$ do
        for $h_\beta \in \mathcal{H}$ do
            $h_f^M \leftarrow \big((h_i)_2^M, h_\beta\big)$
            $P_{0,1,\beta}[0][\langle h_i^M, h_f^M \rangle] \leftarrow p(h_\beta | h_i^M) W(y_\beta | 0, h_\beta)$
            $P_{0,1,\beta}[1][\langle h_i^M, h_f^M \rangle] \leftarrow p(h_\beta | h_i^M) W(y_\beta | 1, h_\beta)$
for $\phi = 1, \dots, N$ do
    computeProb($n$,$\phi$)
    if $\phi \in \mathcal{A}^c$ then                                      // frozen bit
        $B_{n,\phi,1} \leftarrow 0$
    else                                                    // apply equation (7.4)
        $B_{n,\phi,1} \leftarrow \arg\max_{b \in \{0,1\}} \sum_{(h_i^M, h_f^M) \in \mathcal{H}^{2M}} \pi(h_i^M) P_{n,\phi,1}[b][\langle h_i^M, h_f^M \rangle]$
    $L_{out}[\phi] \leftarrow \sum_{h_i^M, h_f^M} \pi(h_i^M) P_{n,\phi,1}[B_{n,\phi,1}][\langle h_i^M, h_f^M \rangle]$
    if $\phi \mod 2 = 0$ then
        updateBitValues($n$,$\phi$)
return $\big(\hat{u}^N = (B_{n,\phi,1})_{\phi=1}^N,\ \hat{x}^N = \hat{u}^N G_n,\ L_{out}\big)$.
```

Theorem 1 hints at a decoding algorithm for Mth-order finite-state Markov channels. Through the recursive computation of the conditional probabilities of $W_\lambda^{(\phi,\beta)}$ using equations (7.6) and (7.7), a decoder can make a decision on U_ϕ by maximizing $\mathsf{P}\{Y_1^N = y_1^N, U_1^{\phi-1} = \hat{u}_1^{\phi-1} \,|\, U_\phi = u_\phi\}$, which can be evaluated for each $u_\phi \in \{0,1\}$ using equation (7.4). Therefore, in the context of a finite-state Markov channel, it makes sense to define $P_{\lambda,\phi,\beta}[b]$ as an array corresponding to the conditional probabilities of $W_\lambda^{(\phi,\beta)}$, where each entry in the array is associated with one tuple $(h_i^M, h_f^M) \in \mathcal{H}^{2M}$. That is, for each $b \in \{0,1\}$ and $(h_i^M, h_f^M) \in \mathcal{H}^{2M}$, we define

$$P_{\lambda,\phi,\beta}[b][\langle h_i^M, h_f^M \rangle] = W_\lambda^{(\phi,\beta)}(y_1^\Lambda, \hat{u}_1^{\phi-1}, h_f^M | b, h_i^M),$$

where $\langle h_i^M, h_f^M \rangle$ is an index corresponding to (h_i^M, h_f^M). It follows that $P_{\lambda,\phi,\beta}[b]$ has a size of $|\mathcal{H}|^{2M}$, and is assumed to be initialized to zeros prior to decoding.

Following this description, Algorithms 1, 2 and 3 show an implementation of the SCTD decoder. Note that Algorithm 3 is identical to [113, Algorithm 4], and is provided here for completion.

Algorithm 2: `computeProb(`λ`,`ϕ`)`

```
Input: Layer $\lambda$, phase $\phi$.
if $\lambda = 0$ then return
$\psi \leftarrow \lceil \phi/2 \rceil$
if $\phi \mod 2 = 1$ then computeProb($\lambda - 1$,$\psi$)
for $\beta = 1, \ldots, 2^{n-\lambda}$ do
    if $\phi \mod 2 = 1$ then                                   // apply equation (7.6)
        for $u \in \{0, 1\}$ do
            for $(h_i^M, h_f^M) \in \mathcal{H}^{2M}$ do
                $P_{\lambda,\phi,\beta}[u][\langle h_i^M, h_f^M \rangle] \leftarrow \sum_{\tilde{h}^M} \sum_{u'} \frac{1}{2}\, P_{\lambda-1,\psi,2\beta-1}[u \oplus u'][\langle h_i^M, \tilde{h}^M \rangle].$
                                        $P_{\lambda-1,\psi,2\beta}[u'][\langle \tilde{h}^M, h_f^M \rangle]$
    else                                                       // apply equation (7.7)
        $u \leftarrow B_{\lambda,\phi-1,\beta}$
        for $u' \in \{0, 1\}$ do
            for $(h_i^M, h_f^M) \in \mathcal{H}^{2M}$ do
                $P_{\lambda,\phi,\beta}[u'][\langle h_i^M, h_f^M \rangle] \leftarrow \sum_{\tilde{h}^M} \frac{1}{2} P_{\lambda-1,\psi,2\beta-1}[u \oplus u'][\langle h_i^M, \tilde{h}^M \rangle].$
                                        $P_{\lambda-1,\psi,2\beta}[u'][\langle \tilde{h}^M, h_f^M \rangle]$
```

The notation $(h_i)_2^M$ in Line 4 of Algorithm 1 means the subvector of h_i^M consisting of the last $M-1$ entries. It is not difficult to see that since the conditional probabilities of $W_\lambda^{(\phi,\beta)}$ need to be computed for each $(h_i^M, h_f^M) \in \mathcal{H}^{2M}$, and each requires a summation over $|\mathcal{H}|^M$ terms $\big($equations (7.6) and (7.7)$\big)$, it follows that the complexity of the algorithm is $\mathcal{O}(|\mathcal{H}|^{3M} N \log N)$. Thus, the algorithm is practical only for low memory order and few-to-moderate number of states.

Note that a *list decoding* version of the algorithm follows almost identically as in the memoryless setting [113]. Instead of estimating each bit as being either zero or one, one can explore both possibilities, while maintaining a maximum of L candidate codewords at each step, where L is the list size. The complexity of the list decoding version of the algorithm is $\mathcal{O}(|\mathcal{H}|^{3M} LN \log N)$.

7.4 Joint Estimation and Decoding

In this section, a joint estimation and decoding scheme at the receiver end is described, composed of two components. The first component is an "estimation-aware" decoder, which leverages the knowledge from channel estimation in the decoding procedure while averaging over the distribution of the channel state. A decoder is presented for each channel model described in Sec-

```
Algorithm 3: updateBitValues($\lambda$,$\phi$)
    Input: Layer $\lambda$, phase $\phi$.
    Require: $\phi$ is even.
    $\psi \leftarrow \lceil \phi/2 \rceil$
    for $\beta = 1, \dots, 2^{n-\lambda}$ do
        $B_{\lambda-1,\psi,2\beta-1} \leftarrow B_{\lambda,\phi-1,\beta} \oplus B_{\lambda,\phi,\beta}$
        $B_{\lambda-1,\psi,2\beta} \quad \leftarrow B_{\lambda,\phi,\beta}$
    if $\psi \mod 2 = 0$ then updateBitValues($\lambda - 1$,$\psi$)
```

tion 7.1. Next, an iterative approach to estimation and decoding is given. The approach uses the output from list decoding to incorporate reliably-decoded bits into subsequent iterations of channel estimation.

7.4.1 Estimation-Aware Decoding

The previous description of SCTD in Section 7.3 assumed no knowledge about the realizations of the channel states. That is, in principle, SCTD can be used to overcome any uncertainty about the channel without the need of transmitting any pilot symbols, as long as the channel is a finite-state Markov channel (Section 7.1.1). However, simulations show that SCTD, as described so far, is far from achieving a reasonable decoding performance in practice, and therefore, the use of pilot symbols is essential. In the following, we first show that SCTD can be modified to allow incorporating channel estimates at the pilot symbol positions into channel decoding. We refer to the modified decoder as an *estimation-aware decoder* for finite-state Markov channels. We then propose a successive cancellation decoder for the Gauss-Markov channel (Section 7.1.2), an essential component of which is using channel estimates in decoding. Finally, by modeling the channel state process in fading channels as a Gauss-Markov process, the proposed decoder for the Gauss-Markov channel can be used to decode polar codes over fading channels (Section 7.1.3). Hereafter, we denote by $\mathcal{P}$ the set of pilot symbol positions and by $\hat{h}_{\mathcal{P}}$ the channel estimates at the pilot positions. For now, we will assume that pilot symbols are inserted periodically *in between* coded symbols to aid in channel estimation (see Figure 7.2)[4]. The channel state realizations at the pilot symbol positions adhere to the correlation properties of the channel state process for each channel model.

Finite-State Markov Channel

The estimation-aware decoder for finite-state Markov channels makes a decision on U_ϕ, for each $1 \leq \phi \leq N$, based on the evaluation of the expression in (7.4) *conditioned* on the channel

[4] The pilot arrangement scheme described in Section 7.5, however, will align pilot symbols with shortened bits of a shortened polar code, i.e., in that case, pilot symbols are part of the codeword.

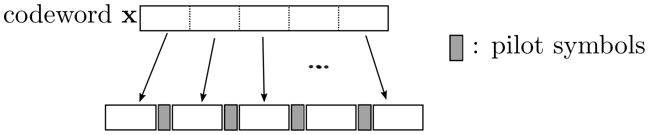


Figure 7.2: Conventional piloting scheme

estimates at the pilot positions. That is, the decoder computes, for each $1 \leq \phi \leq N$ and each $u_\phi \in \{0, 1\}$,

$$\mathsf{P}\{Y_1^N = y_1^N, U_1^{\phi-1} = \hat{u}_1^{\phi-1} | U_\phi = u_\phi, H_\mathcal{P} = \hat{h}_\mathcal{P}\}.$$

The main observation is that conditioning on the channel estimates $\hat{h}_\mathcal{P}$ does not alter the recursive structure of equations (7.6) and (7.7), and the only needed modification to the SCTD algorithm is in the base condition (7.5), i.e., in the computation of conditional probabilities at layer $\lambda = 0$. Notice that we have

$$W_0^{(1,\beta)}(y_\beta, h_\beta | x_\beta, h_{\beta-M}^{\beta-1}, \hat{h}_\mathcal{P}) = p(h_\beta | h_{\beta-M}^{\beta-1}, \hat{h}_\mathcal{P}) W(y_\beta | x_\beta, h_\beta), \tag{7.8}$$

where $p(h_\beta | h_{\beta-M}^{\beta-1}, \hat{h}_\mathcal{P})$ can be efficiently computed using the Markovity of the channel state process. To see this, notice that

$$p(h_\beta | h_{\beta-M}^{\beta-1}, \hat{h}_\mathcal{P}) = \frac{p(\hat{h}_\mathcal{P} | h_{\beta-M+1}^{\beta}) p(h_\beta | h_{\beta-M}^{\beta-1})}{p(\hat{h}_\mathcal{P} | h_{\beta-M}^{\beta-1})},$$

where $p(h_\beta | h_{\beta-M}^{\beta-1})$ are the transition probabilities of the Markov chain, and $p(\hat{h}_\mathcal{P} | h_{\beta-M}^{\beta-1})$, $1 \leq \beta \leq N$, can be computed recursively by noticing that

$$p(\hat{h}_\mathcal{P} | h_{\beta-M}^{\beta-1}) = \sum_{h_\beta} p(h_\beta | h_{\beta-M}^{\beta-1}) p(\hat{h}_\mathcal{P} | h_{\beta-M+1}^{\beta}).$$

Therefore, the estimation-aware decoder substitutes lines 5 and 6 in Algorithm 1 by the corresponding computation designated by equation (7.8). Clearly, this modification does not alter the complexity of the algorithm, which remains at $\mathcal{O}(|\mathcal{H}|^{3M} LN \log N)$. Note that the proposed decoder computes decoding metrics by conditioning on the channel estimates at the pilot symbol positions, while the channel at the data symbol positions is averaged over its Markov distribution. In other words, the proposed decoding algorithm does not use any interpolation between channel estimates.

Gauss-Markov Channel

The SCTD-based decoder described earlier cannot be readily used over a Gauss-Markov channel model (Section 7.1.2). Since the channel states in this model can take any value on the real line (i.e., do not belong to a finite set), equations (7.6) and (7.7) cannot be evaluated for every possible $(h_0, h_f) \in \mathbb{R}^2$. Therefore, a successive cancellation decoder that overcomes this distinction of the Gauss-Markov channel model is described here. We refer to this decoder as the *Gauss-Markov successive cancellation decoder*. The basic assumption of the decoder is that pilot symbols are inserted in between sub-blocks of the codeword that are of size that is a power of 2. Therefore, in what follows, we let $S = 2^s$ denote the spacing between pilot symbols.

As for finite-state channels, the goal for an "estimation-aware" polar decoder over the Gauss-Markov channel is to compute, for each $1 \leq \phi \leq N$,

$$\begin{aligned} &p(\hat{u}_1^{\phi-1}) f\big(y_1^N \,|\, (\hat{u}_1^{\phi-1}, u_\phi), \hat{h}_{\mathcal{P}}\big) \\ &= \sum_{u_{\phi+1}^N} \frac{1}{2^{N-1}} \int f\big(y_1^N, h_{\mathcal{P}^c} \,|\, (\hat{u}_1^{\phi-1}, u_\phi, u_{\phi+1}^N), \hat{h}_{\mathcal{P}}\big)\, \delta h_{\mathcal{P}^c}, \\ &= \sum_{u_{\phi+1}^N} \frac{1}{2^{N-1}} \int f(h_{\mathcal{P}^c} \,|\, \hat{h}_{\mathcal{P}}) \prod_{i=1}^{N} \frac{e^{-\frac{(y_i - h_i \mathrm{s}((u_1^N G_n)_i))^2}{2\sigma_z^2}}}{\sqrt{2\pi\sigma_z^2}}\, \delta h_{\mathcal{P}^c}, \end{aligned} \tag{7.9}$$

where $\hat{u}_1^{\phi-1}$ are the previously decoded bits at the ϕ-th decoding stage and $\hat{h}_{\mathcal{P}}$ are channel estimates at the pilot symbol positions. Notice that for each $u_{\phi+1}^N$, the evaluation of the integral in (7.9) can be done very efficiently due to the Gaussianity of the channel state process. However, the evaluation of (7.9) still requires a summation over $2^{N-\phi}$ terms, which blows up the decoding complexity. The Gauss-Markov successive cancellation decoder overcomes the exponential decoding complexity by making use of the conditioning on the channel estimates at the pilot positions. More precisely, conditioned on the channel estimates $\hat{h}_{\mathcal{P}}$, one can view the marginalization over the joint distribution of the channel states as a product of marginalizations over smaller blocks, each of length equal to the pilot spacing. This follows from the first-order Markovity of the Gauss-Markov channel. When the pilot spacing S is a power of 2, this decomposition is in line with the polar coding recursive structure.

The Gauss-Markov successive cancellation decoder is best understood through an example. Consider a polar code of length $N = 16$. Let pilot symbols be inserted in between sub-blocks of the codeword of length $S = 4$ (see Figure 7.2), and let $\hat{h}_{\mathcal{P}} = (\hat{h}_{p_1}, \hat{h}_{p_2}, \hat{h}_{p_3})$ be the channel estimates at the pilot symbol positions. Let x_1^{16} be any channel input vector. Since the channel state process in a Gauss-Markov channel is first-order Markov, the conditional density $f(y_1^{16} \,|\, x_1^{16}, \hat{h}_{\mathcal{P}})$ can be factorized as

$$f\big(y_1^{16} \,|\, x_1^{16}, \hat{h}_{\mathcal{P}}\big) = f\big(y_1^4 \,|\, x_1^4, \hat{h}_{p_1}\big) f\big(y_5^8 \,|\, x_5^8, \hat{h}_{p_1}, \hat{h}_{p_2}\big) f\big(y_9^{12} \,|\, x_9^{12}, \hat{h}_{p_2}, \hat{h}_{p_3}\big) f\big(y_{13}^{16} \,|\, x_{13}^{16}, \hat{h}_{p_3}\big).$$

This factorization is the essence of the proposed decoder for the Gauss-Markov channel. Note that this equality wouldn't hold without the conditioning on $\hat{h}_{\mathcal{P}}$. Since the pilot spacing is a power of 2, each term in this factorization corresponds to a smaller constituent polar code of length 4. Therefore, one can proceed in the same way as in polar decoding for memoryless channels except that the base case of the recursion would correspond to one term of this factorization.

To this end, recall the notation defined in Section 7.2, in particular the probability variable $P_{\lambda,\phi,\beta}[b]$, where $b \in \{0,1\}$ and the triplet (λ, ϕ, β) satisfies (7.3). In the context of the Gauss-Markov channel, it is meaningful to define $P_{\lambda,\phi,\beta}[b]$ as

$$\begin{aligned} P_{\lambda,\phi,\beta}[b] &= p(\hat{u}_1^{\phi-1}) f\big(y^{\Lambda} \,|\, (\hat{u}_1^{\phi-1}, b), \hat{h}_{\mathcal{P}}\big) \\ &\overset{(a)}{=} p(\hat{u}_1^{\phi-1}) f\big(y^{\Lambda} \,|\, (\hat{u}_1^{\phi-1}, b), \hat{h}_{\text{prev}}, \hat{h}_{\text{next}}\big), \end{aligned} \tag{7.10}$$

where $\hat{h}_{\text{prev}}$ and $\hat{h}_{\text{next}}$ are channel estimates at the previous and next pilot positions respectively, and $S = 2^s$ is the pilot spacing. The equality (a) follows because the channel state process is first-order Markov. Notice that, in this case, $P_{\lambda,\phi,\beta}[b]$ corresponds to a single probability value (i.e., an array of size 1). From (7.9) and (7.10), it follows that, for each $1 \leq \phi \leq N$, we have

$$p(\hat{u}_1^{\phi-1}) f\big(y_1^N \,|\, (\hat{u}_1^{\phi-1}, u_\phi), \hat{h}_{\mathcal{P}}\big) = P_{n,\phi,1}[u_\phi].$$

Algorithms 4 and 5 show a high-level description of the Gauss-Markov successive cancellation decoder. The main distinction compared to the SCTD decoder described previously is that at layer $\lambda = s$, the Gauss-Markov successive cancellation decoder computes the probability pairs at each branch by an exact evaluation through summing over all possible binary vectors in that branch (Lines 9 and 10 in Algorithm 5). In other words, the base case here corresponds to the layer $\lambda = s$, and not $\lambda = 0$. This allows the decoder to use the channel estimates in evaluating the conditional densities.

Notice that given the information vector at layer λ and branch β, $(Y^{\Lambda}, H^{\Lambda})$ is jointly Gaussian. Therefore, for each $u \in \{0,1\}^S$, the evaluation of the integral in Line 9 of Algorithm 5 can be efficiently done since it corresponds to the marginalization of a jointly Gaussian random variable. It follows that the computation done in Line 10 requires summing over at most 2^{S-1} terms, which makes the decoding complexity of the Gauss-Markov successive cancellation decoder in the order of $\mathcal{O}(2^S N \log N)$. Note that the description of the algorithm directly entails a list decoding version of it, analogous to the memoryless setting [113], the complexity of which becomes $\mathcal{O}(2^S L N \log N)$, where L is the list size. The fundamental premise here is that the pilot spacing S is often small, and thus this decoding complexity is still acceptable for practical implementations. Indeed, the pilot spacing for the demodulation reference signal (DMRS) in the 5G New Radio (NR) standards is limited to 4 over various channels, e.g. the physical downlink control channel (PDCCH) over which polar codes are used [120]. In case a large pilot spacing is required, we propose a complexity reduction method in Section 7.6.3, which uses interpolation between channel estimates at certain positions to reduce the decoding complexity.

Fading Channel

To design a decoder for the fading channel model (Section 7.1.3) that takes into account the inherent channel memory, the idea is to model the fading processes as Gauss-Markov processes and re-

Algorithm 4: Gauss-Markov Successive Cancellation Decoder

Input: Received vector y_1^N, channel estimates $\hat{h}_{\mathcal{P}}$ at pilot positions, pilot spacing S, information set $\mathcal{A}$.

Require: $S = 2^s$ for some s.

Output: Decoded information vector $\hat{u}^N$, Decoded codeword $\hat{x}^N$, Array of bit-channel likelihoods L_{out}.

```
for $\phi = 1, \dots, N$ do
    computeProb_GM($y_1^N$,$\hat{h}_{\mathcal{P}}$,$S$,$n$,$\phi$)
    if $\phi \in \mathcal{A}^c$ then                                  // frozen bit
        $B_{n,\phi,1} \leftarrow 0$
    else
        $B_{n,\phi,1} \leftarrow \arg\max_{b \in \{0,1\}} P_{n,\phi,1}[b]$
    if $\phi \mod 2 = 0$ then
        updateBitValues($n$,$\phi$)
$\hat{u}^N = (B_{n,\phi,1})_{\phi=1}^N$; $\hat{x}^N = \hat{u}^N G_n$; $L_{out} = (P_{n,\phi,1}[\hat{u}_\phi])_{\phi=1}^N$
return $\big(\hat{u},\, \hat{x},\, L_{out}\big)$.
```

use the Gauss-Markov successive cancellation decoder. More precisely, since a Gauss-Markov process can be seen as a first-order autoregressive process, we use autoregressive modeling to approximate the underlying fading processes $(I_i)_{i\geq 0}$ and $(Q_i)_{i\geq 0}$ [121]. That is, the processes $(I_i)_{i\geq 0}$ and $(Q_i)_{i\geq 0}$ can be approximated by the first-order autoregressive processes $(\hat{I}_i)_{i\geq 0}$ and $(\hat{Q}_i)_{i\geq 0}$[5], such that, for $1 \leq i \leq N$,

$$\begin{aligned} \hat{I}_i &= \gamma \hat{I}_{i-1} + \hat{V}_i + c, \\ \hat{Q}_i &= \gamma \hat{Q}_{i-1} + \tilde{V}_i + c, \end{aligned} \tag{7.11}$$

where $\gamma \in (0,1)$, $c \in \mathbb{R}$ is a constant, and $\hat{V}_1, \hat{V}_2, \dots$ and $\tilde{V}_1, \tilde{V}_2, \dots$ are i.i.d, zero-mean Gaussian random variables with variance σ_v^2. By solving the Yule-Walker equations [121, Chapter 4], we get that the appropriate parameters of the autoregressive model should be

$$\begin{aligned} \gamma &= \frac{R_I[1]}{R_I[0]}, \\ c &= \big(1-\gamma\big)\frac{\mu}{\sqrt{2}}, \\ \sigma_v^2 &= R_I[0] - \frac{(R_I[1])^2}{R_I[0]}, \end{aligned} \tag{7.12}$$

[5] Note that a higher-order autoregressive process would be a better approximation to the fading process, but an appropriate decoder should be designed in that case.

Algorithm 5: `computeProb_GM`(y_1^N, $\hat{h}_{\mathcal{P}}$, S, λ, ϕ)

Input: Received vector $\mathbf{y}_1^N$, channel estimates $\hat{h}_{\mathcal{P}}$ at pilot positions, pilot spacing S, layer λ, phase ϕ.

```
if $\lambda < s$ then return;
$\psi \leftarrow \lceil \phi/2 \rceil$
if $\lambda = s$ then
    for $\beta = 1, \ldots, 2^{n-\lambda}$ do
        $\hat{u}_1^{\phi-1} \leftarrow (B_{\lambda,\xi,\beta})_{\xi=1}^{\phi-1}$
        prev $\leftarrow$ previous pilot index
        next $\leftarrow$ next pilot index
        for $b \in \{0,1\}$ do
            $P_{\lambda,\phi,\beta}[b] \leftarrow \sum_{\substack{u \in \{0,1\}^S: \\ u_1^{\phi} = (\hat{u}_1^{\phi-1}, b)}} \frac{1}{2^{S-1}} \int f(y_1^{\Lambda}, h_1^{\Lambda} | u, \hat{h}_{\text{prev}}, \hat{h}_{\text{next}}) \delta h_1^{\Lambda}$
else
    if $\phi \mod 2 = 1$ then
        computeProb_GM($y_1^N$, $\hat{h}_{\mathcal{P}}$, $S$, $\lambda - 1$, $\psi$)
    for $\beta = 1, \ldots, 2^{n-\lambda}$ do
        if $\phi \mod 2 = 1$ then
            for $u \in \{0,1\}$ do
                $P_{\lambda,\phi\beta}[u] \leftarrow \sum_{u'} \frac{1}{2} P_{\lambda-1,\psi,2\beta-1}[u \oplus u'].P_{\lambda-1,\psi,2\beta}[u']$
        else
            $u \leftarrow B_{\lambda,\phi-1,\beta}$
            for $u' \in \{0,1\}$ do
                $P_{\lambda,\phi,\beta}[u'] \leftarrow \frac{1}{2} P_{\lambda-1,\psi,2\beta-1}[u \oplus u'].P_{\lambda-1,\psi,2\beta}[u']$
```

where $R_I[.]$ is as defined in equation (7.1). Note that autoregressive modeling of fading channels is not new and has been previously considered in [122].

By using the Gauss-Markov successive cancellation decoder described previously and assuming that the underlying Gaussian processes of the fading process follow the model in (7.11), one can design a decoder for the flat-fading channel. This decoder takes into account the inherent memory in the channel (up to first-order), in contrast to baseline decoders used in practice which completely ignore the channel memory in decoding. Note that a similar approach can be followed for any channel model with memory, provided that the distribution of the channel state process is known to the receiver.

7.4.2 Iterative Estimation and Decoding

The second component of the proposed decoder is an iterative process that alternates between channel estimation and decoding in order to boost the decoding performance. Before we describe the iterative scheme, we state a helpful observation.

From equation (7.4), it can be seen that for $\phi = N$, the polar decoder computes the conditional probability given by

$$\mathsf{P}\{Y_1^N = y_1^N, U_1^{N-1} = \hat{u}_1^{N-1} | U_N = u_N\}.$$

Notice that, for a particular $\hat{u}_N \in \{0, 1\}$, this quantity is proportional to the likelihood of the codeword $\hat{x}_1^N = \hat{u}_1^N G_n$. That is, we have that

$$\mathsf{P}\{Y_1^N = y_1^N, U_1^{N-1} = \hat{u}_1^{N-1} | U_N = u_N\} \propto \mathsf{P}\{Y_1^N = y_1^N | X_1^N = \hat{x}_1^N\}.$$

In other words, if the likelihood of the decoded codeword is of interest, one need not compute it separately; it comes "for free" from the SCTD decoder.

Based on this observation, an iterative process of channel estimation and decoding is proposed. The idea is to utilize the output from list decoding of polar codes to identify bits that have been decoded reliably. Once these bits are identified, another iteration of channel estimation is performed, where the set of reliably decoded bits, along with the original set of pilot symbols, is used for estimation. This gives a larger set of channel estimates, which can boost the decoding performance in subsequent iterations.

What remains is to understand how the output from list decoding can be used to identify reliably-decoded bits. Recall that the output from list decoding is a list of L candidate codewords $\{(\hat{x}_1^N)_1, \ldots, (\hat{x}_1^N)_L\}$, along with the likelihood of each codeword $\mathsf{P}\{Y_1^N = y_1^N | X_1^N = (\hat{x}_1^N)_j\}$, $1 \leq j \leq L$ (based on the previous observation). Using this output, the fraction of candidate codewords that assign each bit to 0 and the fraction of codewords that assign it to 1 are computed, where each candidate codeword is weighted by its channel likelihood. More specifically, after the kth iteration of decoding, we compute, for each $1 \leq i \leq N$ and $b \in \{0, 1\}$,

$$w_i^{(k)}(b) = \frac{\sum\limits_{\substack{j=1, \\ (\hat{x}_i)_j = b}}^{L} \mathsf{P}\{Y_1^N = y_1^N | X_1^N = (\hat{x}_1^N)_j\}}{\sum\limits_{j=1}^{L} \mathsf{P}\{Y_1^N = y_1^N | X_1^N = (\hat{x}_1^N)_j\}},$$

where $(\hat{x}_1^N)_j$ is the jth candidate codeword from list decoding, $(\hat{x}_i)_j$ is its ith bit, and $\mathsf{P}\{Y_1^N = y_1^N | X_1^N = (\hat{x}_1^N)_j\}$ is its channel likelihood. $w_i^{(k)}(b)$ represents the "weighted" fraction of candidate codewords in the list that take the value b in the ith position. If $w_i^{(k)}(b) \geq \delta$ for some threshold parameter $\delta \in (\frac{1}{2}, 1]$, then the ith bit is declared reliable, and its value b is used in a subsequent iteration of channel estimation. That is, if $\mathcal{P}^{(k)}$ is the set of bit positions whose known values are used in the kth iteration of channel estimation, then we have that

$$\mathcal{P}^{(k+1)} = \mathcal{P} \cup \big\{ i : w_i^{(k)}(b) \geq \delta \text{ for some } b \in \{0,1\} \big\},$$

where $\mathcal{P}$ is the set of pilot symbol positions. Needless to mention, at the first iteration (prior to any decoding), the channel estimator uses only the pilot symbols for estimation, i.e., $\mathcal{P}^{(1)} = \mathcal{P}$. Notice that when $\delta = 1$, the bits that are agreed upon by all candidate codewords are deemed reliable and used in subsequent iterations of channel estimation.

The iterative estimation and decoding scheme is summarized in Algorithm 6, where `SCTD_list` and `GaussMarkovDecoder_list` are the list decoding versions of the estimation-aware SCTD decoder and the Gauss-Markov successive cancellation decoder, respectively. The outputs $\mathbf{u}$, $\mathbf{x}$ and $\mathbf{L}_{out}$ of these decoders are all $L \times N$ matrices, where each row corresponds to a candidate codeword. We also assume that the rows are ordered in decreasing order of channel likelihood. Note that, in Line 7 of Algorithm 6, $\mathbf{L}_{out}[j, N]$ is the likelihood of the last bit-channel for the jth candidate codeword, and $\mathbf{x}[j, i]$ is the ith bit of the jth candidate codeword.

7.5 Joint Piloting and Encoding

At the encoder side, we propose a pilot arrangement scheme that uses the special structure of the base matrix G_n to fix some bits *within* the transmitted polar codeword. These bits are used as pilot bit symbols to assist in channel estimation. We show that if the spacing between the pilot bit symbols is chosen to be a power of 2, the low encoding complexity of polar codes can be preserved at $\mathcal{O}(N \log N)$. The scheme can be viewed as a shortening scheme for polar codes, where the shortened bits are aligned with the pilot symbol positions (i.e., shortened bits are actually transmitted), and the knowledge of their values is used at the receiver side for both channel estimation and channel decoding.

Let $\mathcal{P}$ be the desired shortening pattern, i.e., the set of desired shortened bit positions[6]. We would like to design the polar code to have $x_{\mathcal{P}} = 0$ for any codeword x^N. This implies that there are certain parity check constraints that the information vector u^N should satisfy. Therefore, we think of u^N as a codeword chosen from a codebook $\tilde{\mathcal{C}}$ governed by these parity check constraints (see Figure 7.3). If $G_{\mathcal{A}\mathcal{P}}$ denotes the submatrix of G_n consisting of rows with indices in $\mathcal{A}$ and columns with indices in $\mathcal{P}$, then we should have that $x_{\mathcal{P}} = 0 = u_{\mathcal{A}} G_{\mathcal{A}\mathcal{P}}$ (conventionally, frozen bits are set to 0). It is easy to see, therefore, that the parity check matrix of $\tilde{\mathcal{C}}$ is $\tilde{H} = (G_{\mathcal{A}\mathcal{P}})^T$, and thus the code $\tilde{\mathcal{C}}$ is fully defined.

It is not guaranteed though that the encoding of $\tilde{\mathcal{C}}$ can be done efficiently. Nonetheless, we argue in the following that the encoding complexity of $\tilde{\mathcal{C}}$ is linear in the blocklength if the spacing between shortened bits is chosen to be a power of 2. To this end, we assume $S = 2^s$ is the spacing

[6] Since the shortened bits are actually intended to be transmitted as pilot symbols, we think of $\mathcal{P}$ as the set of pilot positions, as before.

Algorithm 6: Iterative Estimation and Decoding

```
Input: Received vector $y_1^N$, pilot set $\mathcal{P}$, pilot spacing $S$, information set $\mathcal{A}$, list size $L$,
  number of iterations $I_{max}$, threshold parameter $\delta$.
Output: Decoded information vector $\hat{\mathbf{u}}$.

for $k = 1, \ldots, I_{max}$ do
    if $k = 1$ then
        // pilot symbols are zero bits
        $\mathcal{P}^{(1)} \leftarrow \mathcal{P}$, $\mathcal{B}^{(1)} \leftarrow \mathbf{0}^{|\mathcal{P}|}$
    else
        // identify reliably decoded bits
        $\mathcal{R} \leftarrow \varnothing$, $\mathcal{B} \leftarrow \varnothing$
        for $i = 1, \ldots, N$, $b \in \{0, 1\}$ do
            $w_i(b) \leftarrow \frac{\sum_{j=1, \mathbf{x}[j,i]=b}^{L} \mathbf{L}_{out}[j,N]}{\sum_{j=1}^{L} \mathbf{L}_{out}[j,N]}$
            if $w_i(b) > \delta$ then
                $\mathcal{R} \leftarrow \mathcal{R} \cup \{i\}$, $\mathcal{B} \leftarrow \mathcal{B} \cup \{b\}$
        $\mathcal{P}^{(k)} \leftarrow \mathcal{P} \cup \mathcal{R}$, $\mathcal{B}^{(k)} \leftarrow \mathbf{0}^{|\mathcal{P}|} \cup \mathcal{B}$
    $\hat{h}_{\mathcal{P}^{(k)}} \leftarrow$ ChannelEstimation($y_1^N$,$\mathcal{P}^{(k)}$,$\mathcal{B}^{(k)}$)
    $[\mathbf{u}, \mathbf{x}, \mathbf{L}_{out}] \leftarrow$ SCTD_list($y_1^N$, $\hat{h}_{\mathcal{P}^{(k)}}$, $S$, $\mathcal{A}$, $L$)
    // Replace with GaussMarkovDecoder_list for Gauss-Markov channel
return $\hat{u}^N = (\mathbf{u}[1,i])_{i=1}^N$.
```

between shortened bits, and define $N_p = \frac{N}{S}$ to be the number of shortened bits (i.e., pilot symbols) within a coded block. Now, consider the set $\mathcal{P}_m$ defined by

$$\mathcal{P}_m = \big\{ m + (k-1)S : \, 1 \leq k \leq N_p \big\}, \quad 1 \leq m \leq S.$$

Given S, $\mathcal{P}_m$ represents a family of shortening patterns, indexed by the position m of the first shortened bit, where the spacing between shortened bits is S. In the following, we will show that for any $1 \leq m \leq S$ such that $S = 2^s$, one can design a shortened polar code with shortening pattern $\mathcal{P}_m$, while maintaining the $\mathcal{O}(N \log N)$ encoding complexity.

Recall from equation (7.2) that the base matrices for polar codes with small blocklengths appear as sub-matrices of the base matrix of codes of larger blocklength. For example, if $N_p = 8$,

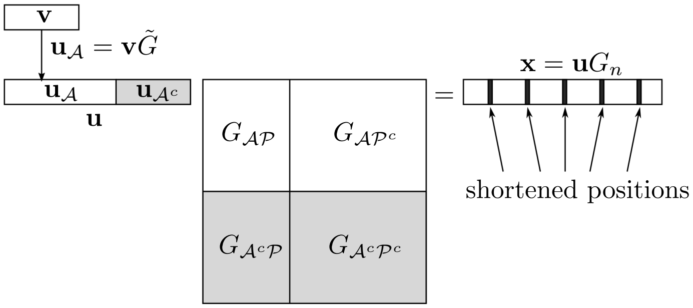

$\mathbf{v}$
$\mathbf{u}_{\mathcal{A}} = \mathbf{v}\tilde{G}$
$\mathbf{u}_{\mathcal{A}}$
$\mathbf{u}_{\mathcal{A}^c}$
$\mathbf{u}$
$G_{\mathcal{A}\mathcal{P}}$
$G_{\mathcal{A}\mathcal{P}^c}$
$G_{\mathcal{A}^c\mathcal{P}}$
$G_{\mathcal{A}^c\mathcal{P}^c}$
$=$
$\mathbf{x} = \mathbf{u}G_n$
shortened positions

Figure 7.3: Shortening of polar codes

then G_n can be written as

$$G_n = \begin{bmatrix} G_s & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ G_s & G_s & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ G_s & \mathbf{0} & G_s & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ G_s & G_s & G_s & G_s & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ G_s & \mathbf{0} & \mathbf{0} & \mathbf{0} & G_s & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ G_s & G_s & \mathbf{0} & \mathbf{0} & G_s & G_s & \mathbf{0} & \mathbf{0} \\ G_s & \mathbf{0} & G_s & \mathbf{0} & G_s & \mathbf{0} & G_s & \mathbf{0} \\ G_s & G_s & G_s & G_s & G_s & G_s & G_s & G_s \end{bmatrix}, \tag{7.13}$$

where each $\mathbf{0}$ here is the $S \times S$ all-zero matrix. Notice that, for the shortening pattern $\mathcal{P}_m$, the parity check constraints in $\tilde{H}$ correspond to columns of G_n that are spaced by S, starting from the m-th column. Due to the lower-triangular structure of G_n, the information vector u^N can be divided into smaller subvectors of length S, where each subvector should satisfy exactly one parity check equation, irrespective of other subvectors. The parity check equation is governed by the mth column of G_s. Therefore, the encoding of $\tilde{\mathcal{C}}$ boils down to the encoding of N_p single-parity-check codes, each with blocklength S. Hence, the encoding complexity of $\tilde{\mathcal{C}}$ is $\mathcal{O}(N_p S) = \mathcal{O}(N)$, and the overall encoding complexity (including polar encoding) is $\mathcal{O}(N \log N)$. Finally, we point out that the shortening scheme can also be used in conjunction with high-order modulation techniques (e.g., QPSK), where consecutive bits are desired to be shortened. In this case, each subvector of u^N should satisfy multiple parity check equations, provided that the spacing between shortened *symbols* is a power of 2. Since these equations are governed by columns of G_s, the encoding complexity can be maintained at $\mathcal{O}(N \log N)$.

This shortening scheme improves on existing shortening schemes (e.g. [123]), since the bits in u^N involved in the parity check constraints do not have to belong to the frozen set, but can be actual information bits. Additionally, the scheme gives flexibility in choosing the index of the first shortened bit, and the only requirement is that the spacing between shortened bits is a power of 2. Note that this construction can be viewed as a concatenation of a parity-check code and a polar code, which can be exploited by a successive cancellation list decoder for early pruning of candidate paths [124]. Also, a benefit of this piloting scheme can arise from implementation considerations, as the scheme allows a fixed-sized decoder to accommodate various pilot densities.

The shortening scheme, albeit being independently interesting, is not the focus of this chapter. In our proposed scheme, we think of the shortened bits as pilot symbols that are transmitted over the channel. We also focus on the special case where the pilot symbol positions (equivalently, the shortened bit positions) are multiples of a power of 2, i.e., when pilot symbols are sent starting from the Sth position. Thus, in the remainder of this chapter, our joint piloting-encoding scheme will consider the set $\mathcal{P}^*$ of pilot symbol positions, where

$$\mathcal{P}^* \triangleq \mathcal{P}_S = \big\{kS : \, 1 \leq k \leq N_p\big\}. \tag{7.14}$$

This particular pilot set is considered since it allows for a lower-complexity design of the polar decoder, particularly when the channel state process is a first-order Markov process (e.g., the Gauss-Markov successive cancellation decoder of Section 7.4.1). The following lemma and corollary highlight the effect of this choice of $\mathcal{P}^*$ on polar encoding.

Lemma 7.5.1. *$G_{\mathcal{P}^{*c}} G_{\mathcal{P}^*} = 0$.*

Proof. Notice that the last column of G_s has a weight of 1, and $G_{S,S} = 1$ (i.e., the 1 appears in the Sth row). Since G_n can be written as the concatenation of G_s matrices and all-zero matrices (take (7.13) as an example), then $G_{\mathcal{P}^{*c}} G_{\mathcal{P}^*} = 0$. □

Corollary 2. *If $\mathcal{P}^*$ is the set of pilot positions, i.e., if $x_{\mathcal{P}^*} = 0$, then $u_{\mathcal{P}^*} = 0$.*

Proof. Notice that $x_{\mathcal{P}^*} = 0 = u_{\mathcal{P}^*} G_{\mathcal{P}^*\mathcal{P}^*} + u_{\mathcal{P}^{*c}} G_{\mathcal{P}^{*c}\mathcal{P}^*}$. Since $G_{\mathcal{P}^{*c}\mathcal{P}^*} = 0$, then $u_{\mathcal{P}^*}$ belongs to the null space of $(G_{\mathcal{P}^*\mathcal{P}^*})^T$. Since $G_{\mathcal{P}^*\mathcal{P}^*}$ is a lower triangular matrix with all diagonal entries being nonzero, then it is full-rank and the null space of $(G_{\mathcal{P}^*\mathcal{P}^*})^T$ includes only the all-zero vector. Thus, $u_{\mathcal{P}^*} = 0$. □

This suggests the following shortened polar code design. Since $u_{\mathcal{P}^*} = 0$ whenever $x_{\mathcal{P}^*} = 0$, then the indices in $\mathcal{P}^*$ can be incorporated into the frozen set of the polar code. That is, the information set $\mathcal{A}$ of the polar code is chosen to be the indices of the bit-channels with the highest symmetric capacity, excluding the ones associated with $\mathcal{P}^*$. Thus, the rate of the underlying polar code is unaltered due to this scheme.

At the decoder side, the knowledge of the pilot bit positions within the codeword can be exploited. More precisely, the introduction of code bits that are known to the decoder creates bits in the polar transformation with conceptually infinite reliability. When used in conjunction with the estimation-aware decoder described in Section 7.4.1, the proposed piloting-encoding scheme can be leveraged by the decoder. In particular, the decoder for finite-state Markov channels would apply a further modification to the base condition (7.8) as follows.

$$W_0^{(1,\beta)}(y_\beta, h_\beta | x_\beta, h_{\beta-M}^{\beta-1}, \hat{h}_{\mathcal{P}^*}) = \begin{cases} 0 & \text{if } \beta \in \mathcal{P}^*, x_\beta = 1, \\ 0 & \text{if } \beta \in \mathcal{P}^*, h_\beta \neq \hat{h}_\beta, \\ W(y_\beta | x_\beta, h_\beta) & \text{if } \beta \in \mathcal{P}^*, h_\beta = \hat{h}_\beta, \\ p(h_\beta | h_{\beta-M}^{\beta-1}, \hat{h}_{\mathcal{P}^*}) W(y_\beta | x_\beta, h_\beta) & \text{otherwise.} \end{cases} \tag{7.15}$$

In principle, this is equivalent to setting an infinite reliability to the measurement received at position $\beta \in \mathcal{P}^*$. On the other hand, when the proposed piloting scheme is used over a Gauss-Markov channel, the Gauss-Markov successive cancellation decoder modifies line 10 of Algorithm 5 by adding the condition $u_S = 0$ in the summation. This exploits the known fact that the last bit within the branch is a pilot bit.

7.6 Simulation Results

7.6.1 Comparison Setup with Separate Estimation and Coding

In this section, the proposed schemes will be compared with a baseline solution of separate estimation and coding, adapted to each of the channel models of Section 7.1. At the transmitter side, this solution consists of inserting pilot symbols periodically *between* coded symbols to assist in channel estimation. At the receiver end, channel estimation is performed at the pilot symbol positions first; then, an interpolator is utilized to get channel estimates at the data symbol positions, and, finally, conventional successive cancellation decoding of polar codes follows assuming the channel estimates are correct (Figure 1.4a).

Notice that the joint estimation and decoding scheme of Sections 7.4 is independent from the joint piloting and encoding scheme presented in Section 7.5. That is, one can use each of the two schemes alongside their corresponding counterpart from the baseline solution of separate estimation and coding, outlined in the previous paragraph. This gives rise to four possible piloting-coding schemes, as shown in Table 7.1. For example, when the joint piloting-encoding scheme is used alongside a separate estimator-decoder, this means that the pilot arrangement pattern of Section 7.5 based on shortened polar codes is used to embed pilot symbols within a codeword, whereas channel estimation, interpolation, and conventional successive cancellation decoding are performed successively at the receiver end.

In the following, the schemes will be compared for the same overall *communication rate*, which is defined as the ratio of the number of information bits to the total number of channel uses. Notice that when the separate piloting-encoding scheme is used (i.e., when pilot symbols are inserted in between coded bits), the communication rate is

$$R_{\text{comm}} = \frac{K_{\text{sep}}}{N + N_p}, \tag{7.16}$$

Table 7.1 Table that shows the considered encoder-decoder pairs, along with the section number where the proposed joint scheme is described.

		Estimation-Decoding	
		Separate	*Joint*
Piloting-Encoding	*Separate*	None (Baseline)	Section 7.4
	Joint	Section 7.5	Section 7.4 + Section 7.5

where N_p is total number of pilot symbols transmitted. On the other hand, when the joint piloting-encoding scheme is used (Section 7.5), the communication rate is

$$R_{\text{comm}} = \frac{K_{\text{joint}}}{N}, \tag{7.17}$$

since pilot symbols are embedded within a polar codeword. Furthermore, the different schemes will be simulated for various pilot densities, and the results will be shown for the pilot spacing that achieves the best block error rate performance for each scheme (and at each SNR value). This allows to compare the whole end-to-end communication schemes for the same overall communication rate. Finally, we point out that for the construction of the polar codes (i.e., the choice of the information set $\mathcal{A}$), we use the construction based on the Gaussian approximation [125], assuming the underlying channel is a binary-input AWGN channel.

For brevity of exposition, we will refer in the sequel to the joint piloting and encoding scheme of Section 7.5 as simply the “joint encoding” scheme, and to the joint estimation and decoding scheme of Section 7.4 as the “joint decoding” scheme. Similar reference will be made to the separate piloting and coding schemes.

7.6.2 Finite-State Markov Channels

The different piloting-coding schemes are simulated over the finite-state Markov channel model described in Section 7.1.1. We consider a second-order Markov channel ($M = 2$), with $\mathcal{H} = \{-1, +1\}$, given by the following input-output relation

$$Y_i = H_i \mathrm{s}(X_i) + Z_i, \quad i = 1, \dots, N,$$

where s and Z_i are as defined in Section 7.1.2, and $H_i \in \mathcal{H}$ is the channel state and follows a second-order Markov chain (i.e., $M = 2$) with transition probabilities

$$\begin{aligned}
\theta_1 &= \mathsf{P}(H_i = +1 \mid H_{i-1} = -1, H_{i-2} = -1) \\
&= \mathsf{P}(H_i = -1 \mid H_{i-1} = +1, H_{i-2} = +1) = 1/64, \\
\theta_2 &= \mathsf{P}(H_i = +1 \mid H_{i-1} = -1, H_{i-2} = +1) \\
&= \mathsf{P}(H_i = -1 \mid H_{i-1} = +1, H_{i-2} = -1) = 1/2.
\end{aligned}$$

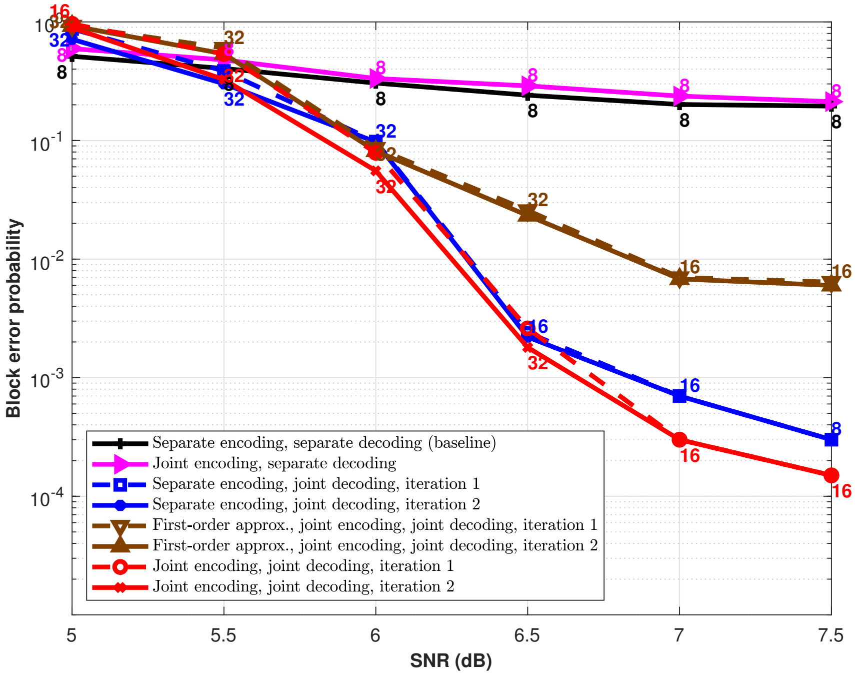


Figure 7.4: Comparison of block error probability performance of the different encoding and decoding schemes using a polar code of blocklength $N = 1024$ over a finite-state second-order Markov channel. The numbers on the plots correspond to the pilot spacing at which the performance was achieved.

The initial state $H_i^M = (H_{-1}, H_0)$ of the Markov chain is chosen according to the stationary distribution of the Markov chain.

Channel estimation at the pilot symbol positions is done through an application of the Viterbi algorithm [126], using the stationary first-order distribution of the Markov chain. For the separate estimator-decoder, channel estimation is followed by interpolation to get estimates at the data symbol positions, which can be done by computing, for $i \notin \mathcal{P}$,

$$\hat{h}_i = \underset{h_i \in \mathcal{H}}{\arg\max}\ p(h_i | \hat{h}_{\mathcal{P}}),$$

where $\hat{h}_{\mathcal{P}}$ is the vector of channel estimates at the pilot symbol positions, and $p(h_i|\hat{h}_{\mathcal{P}})$ can be efficiently computed using the Markovity of the channel state process.

Figure 7.4 shows the block error rate performance of the different schemes over this channel model for an overall communication rate $R_{\text{comm}} = \frac{1}{2}$. The polar code has a block length $N = 1024$, and the decoder uses a list size $L = 32$. The iterative estimation and decoding approach described in Section 7.4.2 uses a threshold parameter $\delta = 1$, and the block error rate performance in the first and second iteration is plotted[7]. At each SNR value, the pilot spacing at which the block error rate

[7] It has been verified through simulations that the performance gain of the iterative approach is negligible beyond the second iteration.

is achieved is shown on the plot of each scheme. Note that this would be the pilot spacing at which the best performance of the scheme is achieved.

In addition to the previously described schemes, Figure 7.4 shows also the performance of a joint decoding scheme that assumes that the channel's underlying Markov chain is its stationary first-order approximation (brown curves), i.e., the decoder in this case only uses the first-order memory in the Markov channel. This essentially reduces the decoding complexity of the joint decoder from $\mathcal{O}(|H|^6 LN \log N)$ to $\mathcal{O}(|H|^3 LN \log N)$. In constrast, the separate estimator and decoder require an application of the Viterbi algorithm for channel estimation and interpolation, followed by an application of the successive cancellation list decoder of polar codes, the total complexity of which becomes $\mathcal{O}(|H|^2 N + LN \log N)$. The small additional complexity incurred by the decoder that considers the first-order approximation of the channel is compensated by a significant performance gain over the separate estimator and decoder (magenta curve). Along the same lines, the comparison with the joint decoder that considers the "true" second-order Markov distribution of the channel state (red curves) highlights the significance of a joint decoder that takes into account the inherent (second-order) memory in the channel, even for the same encoding scheme at the transmitter side. A similar conclusion can be made when comparing the separate and joint decoders for the same separate encoding scheme at the transmitter side (black and blue curves).

On a separate note, the gain from joint decoding is achieved at a larger pilot spacing compared to the schemes of separate decoding. That is, an improved performance can be achieved while sending a smaller number of pilot symbols. As this might seem counter-intuitive at first notice, the reasons are attributed to the following:

- For the joint encoding scheme based on shortened polar codes, the choice of the information set $\mathcal{A}$ is not only based on the bit-channels with the highest symmetric capacities, but also on the set $\mathcal{P}^*$ of shortened bit positions. This significantly affects the performance of the polar code when the pilot spacing is small, where the number of shortened bits is large.
- For the separate encoding scheme on the other hand, a small pilot spacing means a larger number of pilot symbols are transmitted, and thus a larger *code* rate. Recall that the schemes are compared for the same overall communication rate (equations (7.16) and (7.17)).

Finally, by comparing the piloting-encoding schemes for the same decoding strategy (red and blue curves or the magenta and black ones), one can notice that the gain from joint piloting and encoding seems to be small in this case.

7.6.3 Gauss-Markov Channel

The different schemes are simulated for the Gauss-Markov channel model of Section 7.1.2 and compared for an overall communication rate $R_{\text{comm}} = \frac{1}{3}$. The polar code has a block length $N = 1024$, and the decoder uses a list size $L = 32$. The parameters of the Gauss-Markov channel are

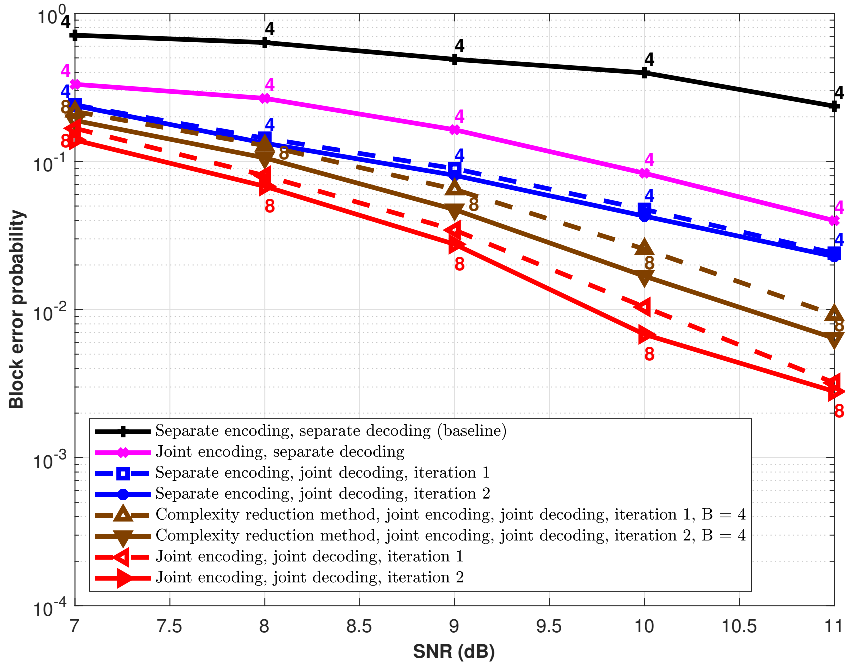


Figure 7.5: Comparison of block error probability performance of the different encoding and decoding schemes over a Gauss-Markov channel with parameters $\eta = 0.99$ and $\sigma_w^2 = 0.0199$. The numbers on the plots correspond to the pilot spacing at which the performance was achieved.

$\eta = 0.99$ and $\sigma_w^2 = 0.0199$. This corresponds to a channel state process whose stationary distribution is $\mathcal{N}(0, 1)$. Figure 7.5 shows the simulation results, where, as before, the number above the plot of each scheme corresponds to the pilot spacing over which the best block error rate performance of the scheme was achieved. Note that for joint decoding, the Gauss-Markov successive cancellation decoder described in Section 7.4.1 is used.

For channel estimation, minimum mean square error (MMSE) estimation is considered. If $\mathcal{P}$ is the set of pilot positions, it is well known that the MMSE estimate of $H_{\mathcal{P}}$ given $Y_{\mathcal{P}}$ is $\hat{H}_{\mathcal{P}} = \mathsf{E}[H_{\mathcal{P}}|Y_{\mathcal{P}}]$. Using the Gaussian assumption of the channel state process, it can be shown that $(H_{\mathcal{P}}, Y_{\mathcal{P}})$ is jointly Gaussian, and $\hat{H}_{\mathcal{P}}$ can be calculated in closed form by

$$\hat{H}_{\mathcal{P}} = \Sigma_{H_{\mathcal{P}}}(\Sigma_{H_{\mathcal{P}}} + \sigma_z^2 I)^{-1} Y_{\mathcal{P}},$$

where $(\Sigma_{H_{\mathcal{P}}})_{ij} = \left(\frac{\sigma_w^2}{1-\eta^2}\right)\eta^{|i-j|S}, \quad 1 \leq i, j \leq |\mathcal{P}|$, is the covariance matrix of $H_{\mathcal{P}}$, and S is the pilot spacing.

For the separate decoding scheme, interpolation follows after channel estimation. At the ith data symbol position, the interpolator minimizes the mean square error of the channel states at the data symbol positions, given the channel estimates at the pilot positions. That is, for $i \notin \mathcal{P}$, the interpolator computes

$$\begin{aligned}\hat{h}_i &= \mathsf{E}[H_i|\hat{H}_{\mathcal{P}} = \hat{h}_{\mathcal{P}}]\\ &\overset{(a)}{=} \mathsf{E}[H_i|H_{\text{prev}} = \hat{h}_{\text{prev}}, H_{\text{next}} = \hat{h}_{\text{next}}],\end{aligned}$$

where $\hat{h}_{\text{prev}}$ and $\hat{h}_{\text{next}}$ are the channel estimates at the previous and next pilot positions, respectively, and (a) follows by the first-order Markovity of the process. Using the Gaussianity of channel states, it is not difficult to show that

$$\hat{h}_i = \frac{\eta^{d_1}(1-\eta^{2d_2})\hat{h}_{\text{next}} + \eta^{d_2}(1-\eta^{2d_1})\hat{h}_{\text{prev}}}{1-\eta^{2S}},$$

where $d_1 = \text{next} - i$ and $d_2 = i - \text{prev}$.

The results in Figure 7.5 clearly show the improvement that both the joint encoding and joint decoding schemes achieve. Note that similar results have been seen for a wide range of other parameters of the Gauss-Markov channel. The superiortiy of the joint encoding scheme compared to the separate scheme is attributed to the smaller code rate (due to embedding pilot symbols within the codeword), as well as the decoder's prior knowledge of the bit values at the shortened positions, which is utilized during decoding (Section 7.5).

Notice that the performance of the joint decoding scheme is achieved at a pilot spacing $S = 8$. Since the complexity of the proposed decoder might be large for this pilot spacing (recall that the complexity of the Gauss-Markov successive cancellation decoder is $\mathcal{O}(2^S LN \log N)$), we present here a simple complexity reduction method that can reduce the decoding complexity. The idea is to use interpolation to get channel estimates at certain intermediate positions between the pilot symbols, and then utilize the proposed decoder given the interpolated estimates. This can be viewed as a middle solution between the separate decoder which uses interpolation to get estimates at all data symbol positions, and the proposed decoder which does not perform any interpolation between the channel estimates. For example, for a pilot spacing of $S = 8$ and a set of pilot symbol positions $\mathcal{P} = \{8, 16, 24, 32, \ldots\}$, one can use the interpolation technique outlined earlier to get estimates of the channel at the positions $\{4, 12, 20, 28, \ldots\}$. Since estimates would be now available at a spacing of $B = 4$, the Gauss-Markov successive cancellation decoder can be utilized, the complexity of which becomes $\mathcal{O}(2^B LN \log N)$. For a small B (e.g., $B = 4$), this complexity is very reasonable and close to that of the successive cancellation list decoder of polar codes. Figure 7.5 shows the performance of this complexity reduction method. Clearly, it still significantly outperforms the separate estimator and decoder, with a comparable decoding complexity.

7.6.4 Fading Channels

The piloting-coding schemes are now applied to the fading channel model described in Section 7.1.3. Figures 7.6, 7.7, and 7.8 show the simulation results for the different coding schemes over a Rayleigh fading channel with normalized Doppler frequency $f_m = 0.06$ (shape parameter $\rho = 0$ and scale parameter $\Omega = 1$), a Rayleigh fading channel with $f_m = 0.1$ (shape parameter $\rho = 0$

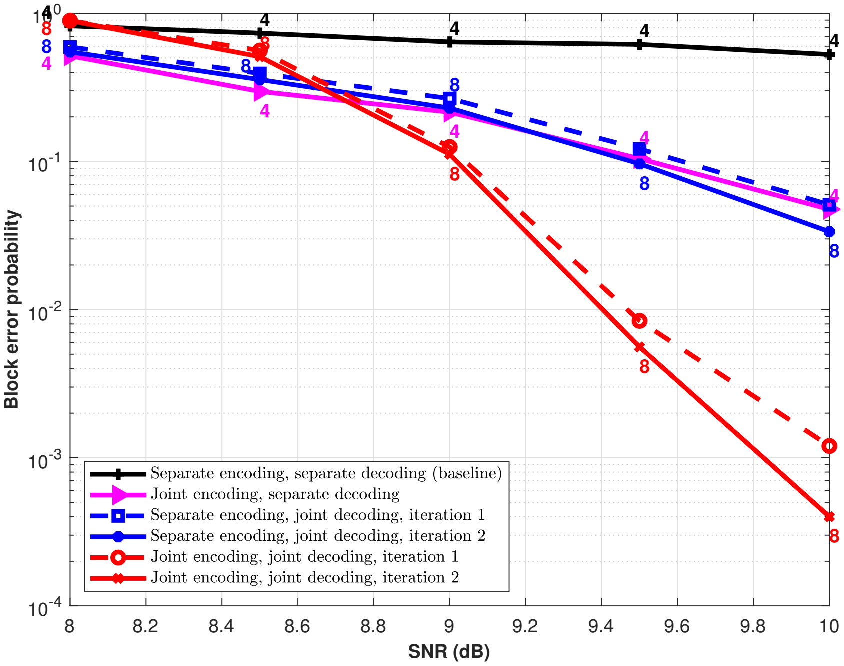


Figure 7.6: Comparison of block error probability performance of the different encoding and decoding schemes over a Rayleigh fading channel with normalized Doppler frequency $f_m = 0.06$. The numbers on the plots correspond to the pilot spacing at which the performance was achieved.

and scale parameter $\Omega = 1$), and a Rician fading channel with $f_m = 0.06$ (shape parameter $\rho = 1$ and scale parameter $\Omega = 1$), respectively. The schemes are compared for an overall communication rate $R_{\text{comm}} = \frac{1}{2}$ using a polar code of block length $N = 1024$ and a decoder with list size $L = 32$.

Linear MMSE (LMMSE) is used for channel estimation at the pilot symbol positions. If $\mathcal{P}$ is the set of pilot positions, the LMMSE estimator of the channel states, given the channel outputs, is

$$\hat{\mathbf{H}}_{\mathcal{P}} = \tilde{\Sigma}_{\mathbf{H}_{\mathcal{P}}}(\tilde{\Sigma}_{\mathbf{H}_{\mathcal{P}}} + \sigma_z^2\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{P}},$$

where $(\tilde{\Sigma}_{\mathbf{H}_{\mathcal{P}}})_{ij} = 2R_I[|i-j|S], \quad 1 \leq i, j \leq |\mathcal{P}|$, is the covariance matrix of $\mathbf{H}_{\mathcal{P}}$, S is the pilot spacing, and $R_I[.]$ is the autocovariance function of the underlying Gaussian processes, as defined in (7.1).

For the separate decoding scheme, interpolation between channel estimates is done through a cubic spline interpolator, which is shown to have a better performance than other interpolators in the literature over the Rayleigh and Rician fading channels (see [127, 128] as examples).

Several observations can be made from the simulation results:

- For the same encoding scheme, the joint estimator-decoder always has a superior block error rate performance compared to the separate estimator-decoder. This highlights again the significance of a decoder that takes into account the inherent channel memory while decoding.

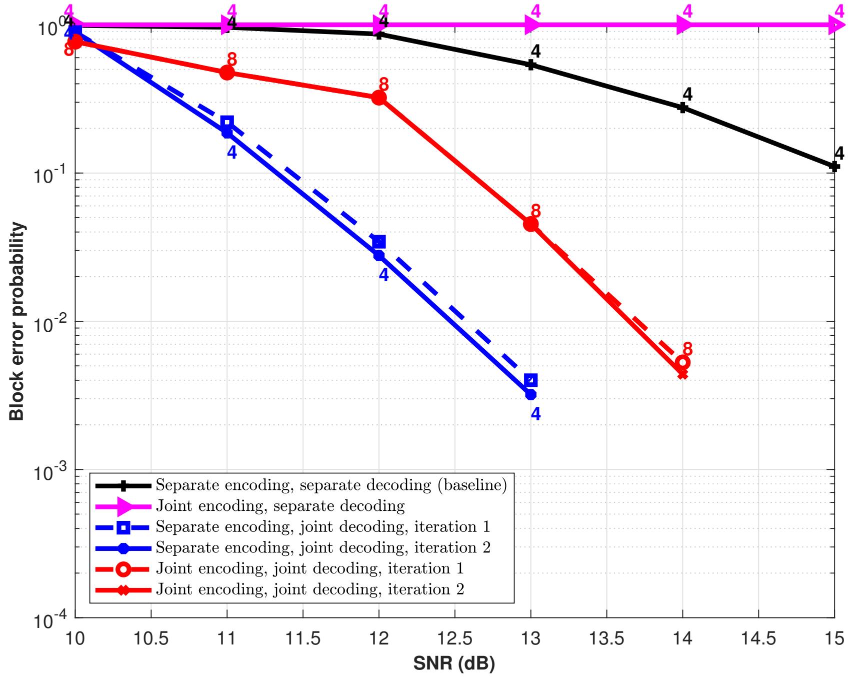


Figure 7.7: Comparison of block error probability performance of the different encoding and decoding schemes over a Rayleigh fading channel with normalized Doppler frequency $f_m = 0.1$. The numbers on the plots correspond to the pilot spacing at which the performance was achieved.

- For the same decoding scheme, comparing the separate and joint encoders reveals that the joint encoding scheme improves the block error rate performance when the normalized Doppler frequency is $f_m = 0.06$ (Figure 7.6 and Figure 7.8), whereas it falls behind for the faster-varying channel (Figure 7.7) with $f_m = 0.1$. We believe that this is the case because fast-varying channels require a small pilot spacing; however, as pointed out before, this limits the design of the shortened polar code of the joint encoding scheme.

- The iterative estimation and decoding scheme of Section 7.4.2 has recorded performance gains in the second iteration of all simulation results of the joint decoding scheme[8].

7.7 Concluding Remarks

The simulation results presented in this chapter are promising and open the door for further study of polar decoding algorithms that take into account the inherent memory in the channel state process. The decoding methods we developed still utilize pilot symbols to track the variations of the channel at the pilot symbol positions, but avoid the naive scheme of interpolation to get channel

[8] Note that even larger gains in the second iteration have been recorded when a smaller overall communication rate was considered. Refer to [129] for a simulation result for $R_{\text{comm}} = \frac{1}{4}$ over a finite-state first-order Markov channel.

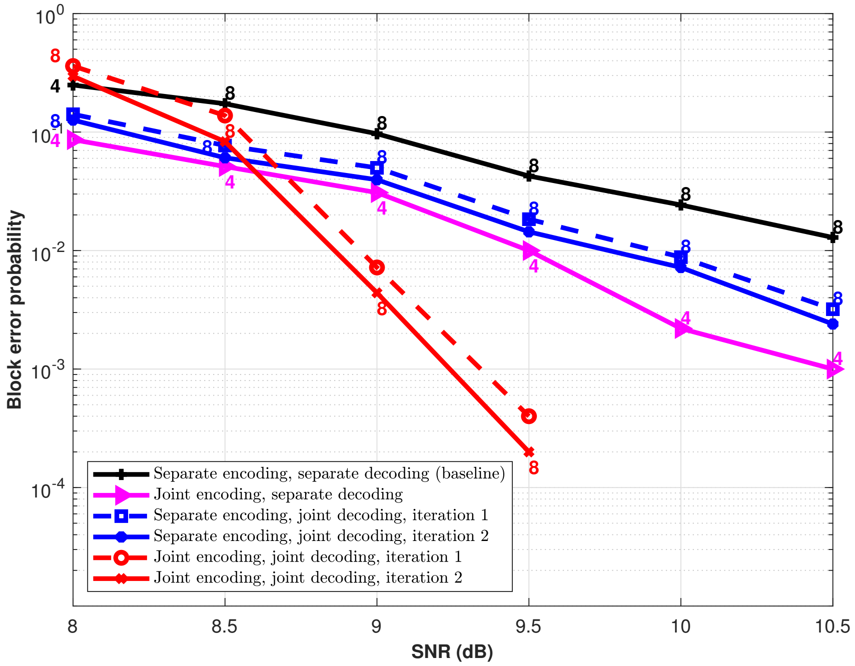


Figure 7.8: Comparison of block error probability performance of the different encoding and decoding schemes over a Rician fading channel with normalized Doppler frequency $f_m = 0.06$. The numbers on the plots correspond to the pilot spacing at which the performance was achieved.

estimates at the data symbol positions prior to decoding. Further, a piloting scheme based on shortened polar codes has shown to improve the decoding performance over several channel models.

As future work, it is of particular interest to reduce the complexity of the proposed decoding algorithms. In this chapter, we have proposed two methods that would allow for such complexity reduction (the first-order approximation for finite-state Markov channels, and the interpolation-based method for the Gauss-Markov successive cancellation decoder). It would be interesting to further improve the decoding complexities, especially for the finite-state Markov channel. Another direction that would be worth exploring is the potential application of the proposed joint estimation and coding scheme to other channel models with memory. For example, an important application is channels with a state that is statistically dependent on the channel input (e.g., the intersymbol interference channel).

Acknowledgement

Chapter 7, in part, is a reprint with permission of the material as it appears in the papers: Nadim Ghaddar, Young-Han Kim, Laurence B. Milstein, Liangping Ma, and Byung K. Yi, “Joint channel estimation and coding over channels with memory using polar codes,” in *IEEE Transactions on Communications*, vol. 69, no. 10, pp. 6575-6589, Oct. 2021, and Nadim Ghaddar, Young-Han

Kim, Laurence B. Milstein, Liangping Ma, and Byung K. Yi, “Joint channel estimation and error correction for finite-state markov channels using polar codes,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2018. The dissertation author was the primary investigator and author of these papers.

Chapter 8

Polar Codes for Multiple Description Coding

In this chapter, we consider the multiple description coding problem. We describe a polar coding scheme that can achieve the entire El Gamal–Cover inner bound for this problem without any time-sharing. In Section 8.1, we give an overview of the literature on polar coding for the multiple description coding problem. In Section 8.2, we describe some polarization results over the multiple access channel. Specifically, we briefly describe the joint polarization technique introduced in [130] and the polarization based on monotone chain rule expansions [131]. Our coding scheme that achieves all points on the dominant face of the El Gamal–Cover rate region without time-sharing is described in Section 8.3. Finally, we conclude in Section 8.4.

8.1 Introduction

Recall the multiple description coding (MDC) problem that was described in Section 3.5. As mentioned before, the optimal rate-distortion region for this problem is not known in general. A number of random-coding-based achievability results have been proposed in the literature [79, 80, 81]. A primitive component in these coding schemes is a joint typicality encoding that generates two descriptions from which we can obtain arbitrarily correlated reconstructions. While producing the best known achievability results, joint typicality encoding is nontrivial to implement in a time/space efficient manner, as it involves multiple codeword-sequence detection at the core of its operation. In this chapter, we look into implementing joint typicality encoding at low-complexity using polar codes [6].

Using polar codes to implement joint typicality encoding was previously considered by [132] for the MDC problem. The coding scheme in [132] targets a corner point of the El Gamal–Cover

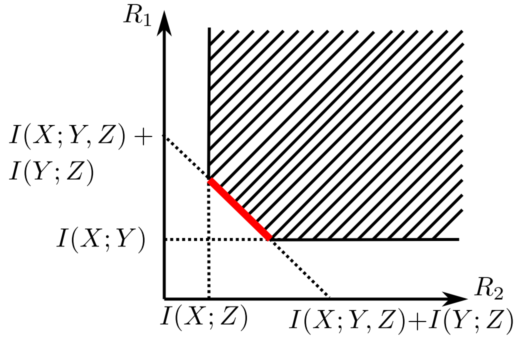


Figure 8.1: El Gamal–Cover inner bound for a fixed pmf $p(y,z\,|\,x)$. The dominant face of this region is highlighted in red.

rate region [79] for this problem (see equation (3.8)) and is based on the *joint polarization* technique, which was initially proposed in [130] in the context of multiple access channels (MAC's). The polar coding scheme in [132] is insufficient to achieve the entire El Gamal–Cover inner bound for the MDC problem *without time-sharing*. Therefore, in this chapter, we explore polar coding schemes that can implement joint typicality encoding in full generality. By exploring Arıkan's polarization technique based on chain rule expansions of the mutual information [131], which was initially proposed in the context of Slepian–Wolf coding, we show that the proposed polar coding scheme can achieve the entire EGC inner bound without any time-sharing.

To this end, we will focus on the El Gamal–Cover (EGC) inner bound for the MDC problem. An equivalent form of this bound is given as follows [133].

Theorem 3 (El Gamal–Cover inner bound [79, 133])**.** *A rate pair (R_1, R_2) is achievable for the multiple description problem with distortion triple (D_0, D_1, D_2) if*

$$\begin{aligned} R_1 &\geq I(X;Y), \\ R_2 &\geq I(X;Z), \\ R_1 + R_2 &\geq I(X;Y,Z) + I(Y;Z) \end{aligned} \tag{8.1}$$

for some conditional pmf $p(y,z|x)$ and some deterministic mappings $\phi_0\colon \mathcal{Y}\times\mathcal{Z}\to\hat{\mathcal{X}}_0$, $\phi_1\colon \mathcal{Y}\to\hat{\mathcal{X}}_1$, and $\phi_2\colon \mathcal{Z}\to\hat{\mathcal{X}}_2$ such that $D_0 \geq \mathbb{E}[d_0(X,\phi_0(Y,Z))]$, $D_1 \geq \mathbb{E}[d_1(X,\phi_1(Y))]$, and $D_2 \geq \mathbb{E}[d_2(X,\phi_2(Z))]$.

Here Y and Z can be seen as the two descriptions representing the source X, and functions $\phi_j, j = 0,1,2$, are the reconstruction functions based on the available descriptions at each decoder. For a fixed $p(y,z|x)$ and functions $\phi_j, j = 0,1,2$, the subset of achievable rate pairs (R_1, R_2) that satisfy $R_1 + R_2 = I(X;Y,Z) + I(Y;Z)$ is called the *dominant face* of rate–distortion region, as illustrated in the red line of Figure 8.1.

8.2 Polarization over Multiple Access Channels

8.2.1 Joint Polarization Technique

In this section, we review the joint polarization technique introduced in [130]. Consider a two-user MAC $p(y|x_1, x_2)$, as defined previously in Section 4.3, and recall the achievable MAC rate region (equation (4.8)),

$$\begin{aligned} R_1 &\leq I(X_1; Y|X_2), \\ R_2 &\leq I(X_2; Y|X_1), \\ R_1 + R_2 &\leq I(X_1, X_2; Y). \end{aligned}$$

for some fixed input pmf $p(x_1)p(x_2)$. Similar to the MDC region, the rate pairs (R_1, R_2) that satisfy $R_1 + R_2 = I(X_1, X_2; Y)$ in the above region is called the *dominant face* of the MAC region.

In [130], a technique, termed *joint polarization*, is proposed. For $N = 2^n$, define the polar transform: let $U^N = X_1^N G_N$ and $V^N = X_2^N G_N$ for $G_N = B_N F_N$, where B_N is the bit-reversal matrix defined in [6] and $F_N = F_1^{\otimes n}$ is the n-th power Kronecker product of the matrix

$$F_1 \triangleq \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Similar to the single-user case, two independent uses of W are transformed into two MACs W^- and W^+, as depicted in Figure 8.2. Consecutively, by applying n levels of this transformation, $N = 2^n$ different MAC channels are created. For $i \in [N]$, consider the mutual information triple

$$\begin{aligned} &\big(I(U_i; Y^N|U^{i-1}, V^i), \\ &I(V_i; Y^N|U^i, V^{i-1}), \\ &I(U_i, V_i; Y^N|U^{i-1}, V^{i-1})\big). \end{aligned}$$

It is shown in [130] that, as n goes to infinity, the mutual information triple approaches one of the five points in following set with high probability

$$(0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 1), (1, 1, 2).$$

In other words, five extremal channels are approached for a two-user MAC, as compared to two in the single-user case (either noise-free or pure noise channels). The point (0,0,0) correspond to the case when the output provides no information about any of the two inputs. The points (1,0,1) and (0,1,1) correspond to the cases when the output provides full information about one of the inputs but provides no information about the other input. The point (1,1,2) corresponds to the case when the output provides full information about both inputs. Finally, the point (1,1,1) is a pure *contention* channel: if any of the two users communicates at zero rate, then the output will provide full information about the other user's input.

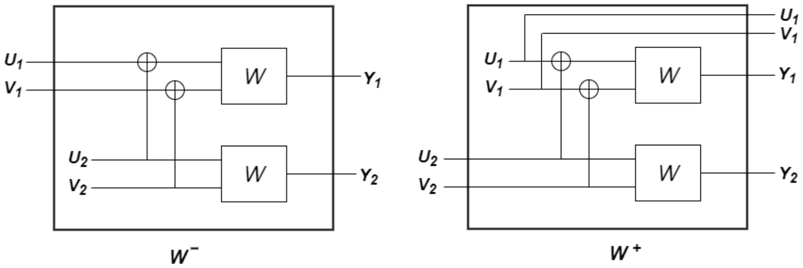


Figure 8.2: Channel splitting operation for two uses of a two-user MAC under the technique of joint polarization

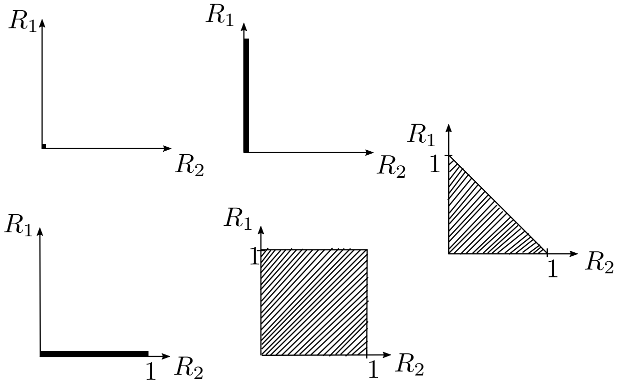


Figure 8.3: Five extremal channels for MAC.

Coding over these extremal channels is simple: either send an information bit or freeze the bit to some known value by the decoder, depending on the corresponding extremal channel. For the (1,1,1) channel, simply assign an information bit to one of the users arbitrarily while freezing the other user's input. It is shown in [130] that the constructed polar code can achieve some rate point on the dominant face of the MAC region.

Note however that the above joint polarization technique does not achieve *any* rate point in the MAC region. This is because joint polarization only considers a single order in expanding (U^N, V^N) in the mutual information term, namely

$$I(U^N, V^N; Y^N) = \sum_{i=1}^{N} I(U_i, V_i; Y^N | U^{i-1}, V^{i-1}).$$

Thus, the symbols are decoded successively in the order $(U_1, V_1), (U_2, V_2), \cdots, (U_N, V_N)$. By exploiting different decoding orders, one can achieve different points on the dominant face of the MAC region. This motivates the next part.

8.2.2 Polarization Based on Monotone Chain Rules

Now we give an overview of the scheme proposed by Arıkan in [131] which achieves any point on the dominant face of the MAC region. The scheme is based on exploiting all possible expansions of the joint mutual information between the channel inputs and outputs. Let $U^N = X_1^N G_N$ and $V^N = X_2^N G_N$ be the random variables induced by the polar transform of X_1^N and X_2^N respectively.

Consider now the expansions of the form

$$\sum_{i=1}^{2N} I(S_i; Y^N | S^{i-1}),$$

where $S^{2N} = (S_1, \ldots, S_{2N})$ is a *monotone* permutation of (U^N, V^N), i.e., S^{2N} is a permutation where the relative order of the elements of both U^N and V^N is preserved. For example, $(U_1, U_2, U_3, U_4, V_1, V_2, V_3, V_4)$ and $(U_1, V_1, U_2, V_2, U_3, V_3, U_4, V_4)$ are monotone permutations of (U^4, V^4), but $(U_1, U_2, U_3, U_4, V_1, V_4, V_3, V_2)$ is not because the order on (V_1, V_2, V_3, V_4) is not preserved. S^{2N} is assumed to be known by both transmitters and the receiver. Also, let $\mathcal{S}_U$ and $\mathcal{S}_V$ denote the set of indices of S^{2N} such that $S_i = U_k$ and $S_i = V_k$ respectively and define the rates

$$R_1 = \frac{1}{N} \sum_{i \in \mathcal{S}_U} I(S_i; Y^N | S^{i-1}),$$
$$R_2 = \frac{1}{N} \sum_{i \in \mathcal{S}_V} I(S_i; Y^N | S^{i-1}).$$

The main contribution of [131] is that the pair (R_1, R_2) can approach any rate pair on the dominant face of the capacity region by selecting a valid permutation S^{2N} and that the mutual informations $I(S_i; Y^N | S^{i-1})$ become polarized with increasing N (i.e., asymptotically approach either 0 or 1). Also, it is shown that permutations of the form $S^{2N} = (U^i, V^N, U_{i+1}^N)$ are sufficient to guarantee this result. Namely, the following theorem holds:

Theorem 4 ([131])**.** *For each $\epsilon > 0$ and $\beta < 1/2$, and rate pair (R_x, R_y) on the dominant face of the MAC region, there exists an N and a permutation $S^{2N} = (U^i, V^N, U_{i+1}^N)$ for some i where*

(i) $|R_1 - R_x| < \epsilon$ *and* $|R_2 - R_y| < \epsilon$

(ii)

$$\frac{N - |\mathcal{F}_1|}{N} > R_1 - \epsilon \quad \textit{and} \quad \frac{N - |\mathcal{F}_2|}{N} > R_2 - \epsilon,$$

where

$$\mathcal{F}_1 = \{1 \leq i \leq 2N : i \in \mathcal{S}_U, I(S_i; Y^N | S^{i-1}) < 2^{-N^\beta}\},$$
$$\mathcal{F}_2 = \{1 \leq i \leq 2N : i \in \mathcal{S}_V, I(S_i; Y^N | S^{i-1}) < 2^{-N^\beta}\}.$$

8.3 Polar Codes for the MDC problem

8.3.1 MDC-MAC Duality

Polar codes have been considered in [132] for the multiple description coding problem. Based on a joint polarization technique similar to [130], a polar coding scheme is shown to achieve one point on the dominant face of the EGC inner bound. The key point in this result is an MDC-MAC duality, that we outline in this section. Consider again the MDC problem described with source X and descriptions (Y, Z). Let the random variables (X, Y, Z) be defined over $\mathbb{F}_q$, where Y and Z are each uniformly distributed over $\mathbb{F}_q$. Such an assumption is justified since any random variable can be approximated arbitrarily well by another that is uniformly distributed over a sufficiently large alphabet size q through a deterministic mapping. Given a probability mass function $p(x, y, z)$ for the MDC problem, the induced conditional distribution $p(x|y, z)$ can be viewed as a MAC with inputs (Y, Z) and output X. Recall that for fixed MAC distribution $p(x|y, z)$ where $p_{Y,Z}(y, z) = p_Y(y)p_Z(z)$, the rate region is the set of non-negative rate pairs (R_1, R_2) such that

$$\begin{aligned} R_1 &\leq I(X, Z; Y), \\ R_2 &\leq I(X, Y; Z), \\ R_1 + R_2 &\leq I(X; Y, Z). \end{aligned}$$

In comparison with the MDC rate region in (8.1), it can be seen that the two regions will have the same *sum-rate* if Y and Z are independent. Figure 8.4 shows the two regions in the case that Y and Z are independent. This is not necessarily true for the MDC problem in general, where Y and Z are the two descriptions of the source. Nevertheless, the independence of the two descriptions can be achieved via a "dithering step" [132]: Let $Z^{'}$ be a random variable uniformly distributed over $\mathbb{F}_q$ and independent of (X, Y, Z). Define $\widetilde{X} = (X, Z')$, $\widetilde{Y} = Y$ and $\widetilde{Z} = Z \oplus Z'$. Then clearly $\widetilde{Y}$ and $\widetilde{Z}$ are independent and the following holds:

$$\begin{aligned} I(\widetilde{X}; \widetilde{Y}) &= I(X; Y), \\ I(\widetilde{X}, \widetilde{Z}; \widetilde{Y}) &= I(X, Z; Y), \\ I(\widetilde{X}; \widetilde{Z}) &= I(X; Z), \\ I(\widetilde{X}, \widetilde{Y}; \widetilde{Z}) &= I(X, Y; Z). \end{aligned} \tag{8.2}$$

Also, we have that

$$\begin{aligned} I(\widetilde{X}; \widetilde{Y}, \widetilde{Z}) + I(\widetilde{Y}; \widetilde{Z}) &= I(\widetilde{X}; \widetilde{Y}, \widetilde{Z}) \\ &= I(\widetilde{X}; \widetilde{Y}) + I(\widetilde{X}; \widetilde{Z}|\widetilde{Y}) + I(\widetilde{Y}; \widetilde{Z}) \\ &= I(\widetilde{X}; \widetilde{Y}) + I(\widetilde{X}, \widetilde{Y}; \widetilde{Z}) \\ &= I(X; Y) + I(X, Y; Z) \\ &= I(X; Y, Z) + I(Y; Z). \end{aligned}$$

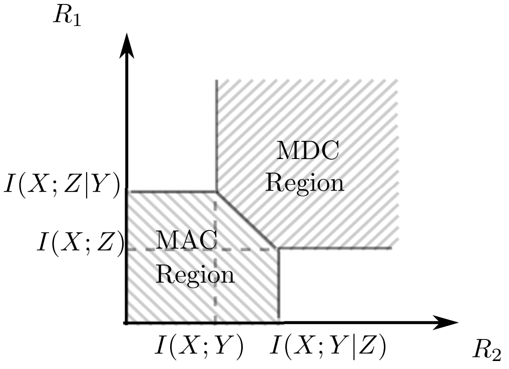


Figure 8.4: MDC-MAC duality.

Hence, the EGC rate region does not change under this transformation.

Following this, it can be shown that, similar to a MAC, the sum-rate on the dominant face of the EGC region is preserved under the polar transformation. Also, we have that

$$\begin{aligned} I(\widetilde{X};\widetilde{Y},\widetilde{Z}) &= I(X;Y,Z) + I(Y;Z) \\ &= I(X;Y) + I(X,Y;Z) \\ &= I(\widetilde{X};\widetilde{Y}) + I(X,Y;Z) \end{aligned} \tag{8.3}$$

and, similarly,

$$I(\widetilde{X};\widetilde{Y},\widetilde{Z}) = I(\widetilde{X};\widetilde{Z}) + I(X,Z;Y). \tag{8.4}$$

Hence, it follows from the results of [130] for the polarization of $I(X,Z;Y)$ and $I(X,Y;Z)$ that $I(\widetilde{X};\widetilde{Y})$ and $I(\widetilde{X};\widetilde{Z})$ also polarize. Hence, five extremal regions are approached asymptotically for the MDC problem as well. On each of these regions, encoding follows naturally from the MAC case. Figure 8.5 shows the extremal regions for the MDC problem, where $\Delta = \log_2 q$.

It is also shown in [132] that the induced distribution from the polar transformation satisfies the distortion constraints, and consequently one rate pair on the dominant face of the EGC region can be achieved asymptotically using this scheme.

In what follows, we argue that the whole dominant face of the EGC region can be achieved, if we consider different monotone chain rule expansions of the mutual information (i.e. similar to the approach of [131]), while still satisfying the distortion constraints imposed by the problem.

8.3.2 Proposed Scheme

We now describe the proposed polar coding scheme for the MDC problem. Let (X_i, Y_i, Z_i) be N i.i.d. copies of $(X, Y, Z), i = 1, \ldots, N$ distributed according to $p_{X,Y,Z}(x, y, z)$ where $N = 2^k$ is

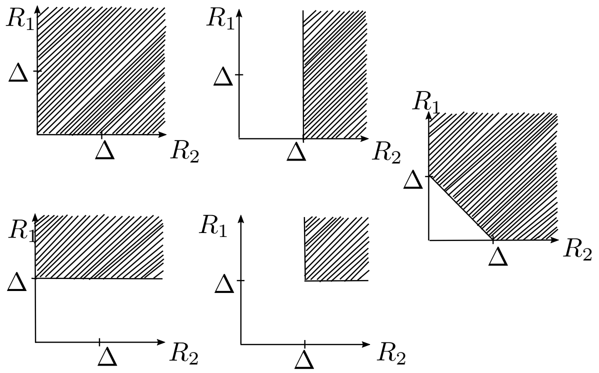


Figure 8.5: Five extremal channels for the MDC problem.

the code block length. As indicated before, we assume that random variables Y and Z are uniformly distributed over $\mathbb{F}_q$. The induced distribution from the polar transformation P_{X^N,U^N,V^N} is

$$\begin{aligned} & p_{X^N,U^N,V^N}(x^N,u^N,v^N) \\ & = \sum_{y^N}\sum_{z^N}\prod_{i=1}^{N} p_{X,Y,Z}(x_i,y_i,z_i)p(u^N|y^N)p(v^N|z^N), \end{aligned}$$

where

$$\begin{aligned} p(u^N|y^N) &= \mathbb{1}_{\{u^N=y^N G_N\}}, \\ p(v^N|z^N) &= \mathbb{1}_{\{v^N=z^N G_N\}}. \end{aligned}$$

The polar coding scheme is defined by two parameters (S^{2N},β), where S^{2N} is a monotone chain rule for $U^N V^N$, as defined in [131], and β is a threshold parameter where $0<\beta<1/2$. We also consider two subsets $\mathcal{F}_1$ and $\mathcal{F}_2$ of $\{1,2,\ldots,2N\}$, defined as follows:

$$\begin{aligned} \mathcal{F}_1 &= \{1 \le i \le 2N : i \in \mathcal{S}_U, I(S_i;Y^N|S^{i-1}) < 2^{-N^\beta}\}, \\ \mathcal{F}_2 &= \{1 \le i \le 2N : i \in \mathcal{S}_V, I(S_i;Y^N|S^{i-1}) < 2^{-N^\beta}\}. \end{aligned} \tag{8.5}$$

Encoding: For each $i \in \mathcal{F}_1$ or $i \in \mathcal{F}_2$, generate s_i at random over $\mathbb{F}_q$. The frozen symbols are generated once and informed to both the encoder and the decoder, and are fixed throughout the communication. If $i \notin \mathcal{F}_1$ and $i \notin \mathcal{F}_2$, then s_i will take value $a \in \mathbb{F}_q$ with probability given by $\frac{P_{X^N,S^i}(x^N,(s^{i-1},a))}{P_{X^N,S^{i-1}}(x^N,s^{i-1})}$. So description 1 will be $s_{\mathcal{F}_1^C} = \{s_i : i \notin \mathcal{F}_1, i \in \mathcal{S}_U\}$ and description 2 will be $s_{\mathcal{F}_2^C} = \{s_i : i \notin \mathcal{F}_2, i \in \mathcal{S}_V\}$.

Decoding: Decoder 1 will form u^N using the first description and the known frozen symbols of u. Then it will generate $y^N = u^N B_N G_N$ and apply ϕ_1 to each symbol of y^N and the output will be the reconstruction $\hat{X}_1^N$. Similarly decoder 2 will form v^N using the first description and the known frozen symbols of v. Then it will generate $z^N = v^N B_N G_N$, and apply ϕ_2 to each symbol of z^N as reconstruction $\hat{X}_2^N$. Decoder 0 forms u^N and v^N, generates y^N and z^N and applies ϕ_0 to $(y_i, z_i), i = 1, 2, \cdots, N$ and gets reconstruction $\hat{X}_0^N$.

Analysis of average distortion: Now we show that this scheme satisfies the distortion constraints. Note that the encoding procedure described above only approximates $p(x^N, s^{2N})$. We want to show that the excess distortion due to this approximation is bounded. First define the distribution $\hat{p}$ to be the distribution induced by our encoding procedure. It follows that

$$\hat{p}(s_i|s^{i-1}, x^N) = \begin{cases} \frac{1}{q} & \text{if } i \in \mathcal{F}_1 \text{ or } i \in \mathcal{F}_2 \\ p(s_i|s^{i-1}, x^N) & \text{otherwise.} \end{cases}$$

Let $\hat{D}_j, j = 0, 1, 2$ be the expected distortions under distribution $\hat{p}$, while $D_j^*, j = 0, 1, 2$ be the expected distortions under p. The following theorem states that the difference between the two distortions can be made arbitrarily small.

Theorem 5. *For any $\epsilon > 0$, there exists a large enough blocklength N such that $|\hat{D}_j - D_j^*| < \epsilon, j = 0, 1, 2$.*

Proof. First note that

$$\begin{aligned} |\hat{D}_1 - D_1^*| &= \Big|\mathbb{E}_{\hat{p}}[d_1^{(N)}(X^N, \phi_1^{(N)}(U^N B_N G_N))] - \mathbb{E}_p[d_1^{(N)}(X^N, \phi_1^{(N)}(U^N B_N G_N))]\Big| \\ &\le d_{\max} \sum_{x^N, s^{2N}} \big|\hat{p}(x^N, s^{2N}) - p(x^N, s^{2N})\big|. \end{aligned}$$

Similarly,

$$|\hat{D}_2 - D_2^*| \le d_{\max} \sum_{x^N, s^{2N}} \big|\hat{p}(x^N, s^{2N}) - p(x^N, s^{2N})\big|,$$

$$|\hat{D}_0 - D_0^*| \le d_{\max} \sum_{x^N, s^{2N}} \big|\hat{p}(x^N, s^{2N}) - p(x^N, s^{2N})\big|.$$

Now we have

$$\begin{aligned} &\sum_{x^N, s^{2N}} \big|\hat{p}(x^N, s^{2N}) - p(x^N, s^{2N})\big| = \sum_{x^N, s^{2N}} p(x^N)\big|\hat{p}(s^{2N}|x^N) - p(s^{2N}|x^N)\big| \\ &= \sum_{x^N, s^{2N}} p(x^N)\big| \prod_{i=1}^{2N} \hat{p}(s_i|x^N, s^{i-1}) - \prod_{i=1}^{2N} p(s_i|x^N, s^{i-1})\big| \\ &\overset{(a)}{=} \sum_{x^N, s^{2N}} p(x^N)\Bigg| \sum_{i=1}^{2N} (p(s_i|x^N, s^{i-1}) - \hat{p}(s_i|x^N, s^{i-1}))\Big(\prod_{j=1}^{i-1} p(s_j|x^N, s^{j-1}) \prod_{j=i+1}^{N} \hat{p}(s_j i x^N, s^{ji-})\Big)\Bigg| \\ &\le \sum_{i=1}^{2N} \sum_{x^N, s^{2N}} p(x^N)\Bigg|(p(s_i|x^N, s^{i-1}) - \hat{p}(s_i|x^N, s^{i-1}))\Big(\prod_{j=1}^{i-1} p(s_j|x^N, s^{j-1}) \prod_{j=i+1}^{N} \hat{p}(s_j i x^N, s^{ji-})\Big)\Bigg| \\ &= \sum_{i=1}^{2N} \sum_{x^N, s^i} p(s^{i-1}, x^N)\big|(p(s_i|x^N, s^{i-1}) - \hat{p}(s_i|x^N, s^{i-1}))\big|, \end{aligned}$$

where (a) follows since

$$\prod_{i=1}^{K} A_i - \prod_{i=1}^{K} B_i = \sum_{i=1}^{K}(A_i - B_i) \prod_{j=1}^{i-1} A_j \prod_{j=i+1}^{K} B_j.$$

Therefore we have for $j = 0, 1, 2,$

$$|\hat{D}_j - D_j^*| \leq d_{\max} \sum_{i=1}^{2N} E_i,$$

where $E_i = \sum_{s_i=0}^{q-1} \mathbb{E}_p[|p(s_i|X^N, S^{i-1}) - \hat{p}(s_i|X^N, S^{i-1})|]$. We have two cases:

1) Case 1: $i \in \mathcal{F}_1$ or $i \in \mathcal{F}_2$

$$\begin{aligned} E_i &= \sum_{s_i=0}^{q-1} \mathbb{E}_p\Big[|p(s_i|X^N, S^{i-1}) - \frac{1}{q}|\Big] \\ &\overset{(a)}{\leq} \sqrt{(2\log^{-1} e) I(X^N, S^{i-1}; S_i)} \\ &\leq \sqrt{(2\log^{-1} e)\delta}, \end{aligned}$$

where (a) follows from Pinsker's inequality and δ is a threshold parameter such that $\delta = \mathcal{O}(2^{-N^{\beta}})$.

2) Case 2: $i \notin \mathcal{F}_1$ and $i \notin \mathcal{F}_2$. In this case, clearly $E_i = 0$.

Combining these two cases, we get that for $j = 0, 1, 2,$

$$\begin{aligned} |\hat{D}_j - D_j^*| &\leq d_{\max}(2N)\sqrt{(2\log^{-1} e)\delta} \\ &= O(2^{-N^{\beta'}}), \end{aligned}$$

for any $\beta' \in (0, \beta)$.

□

Achieving the entire EGC rate region: We will now show that our scheme can approach any point on the dominant face of the EGC region arbitrarily closely. First, define the rate pairs:

$$\begin{aligned} R_1 &= \frac{1}{N} \sum_{i \in \mathcal{S}_U} I(S_i; X^N | S^{i-1}), \\ R_2 &= \frac{1}{N} \sum_{i \in \mathcal{S}_V} I(S_i; X^N | S^{i-1}). \end{aligned}$$

In an approach similar to Theorem 2 in [131], we can show that the terms $I(S_i; X^N|S^{i-1})$ asymptotically approach 0 or 1, and that

$$\frac{N - |\mathcal{F}_1|}{N} \to R_1 \quad and \quad \frac{N - |\mathcal{F}_2|}{N} \to R_2$$

where $\mathcal{F}_1$ and $\mathcal{F}_2$ are as defined in (8.5). Also, adapting an analysis similar to [131], we can directly see that if (U^N, V^N) is a pair obtained from (Y^N, Z^N) via the polar transformation defined in

Section 8.2.1, then any monotone chain rule expansion on U^N, V^N should satisfy

$$\begin{aligned} R_1 &\geq I(X;Y), \\ R_2 &\geq I(X;Z), \\ R_1 + R_2 &= I(X;Y,Z). \end{aligned}$$

In general, corner points of this region are not necessarily the corner points of the EGC rate region. However, following the "dithering step" defined in section 8.3.1, we know that:

$$I(\widetilde{X};\widetilde{Y},\widetilde{Z}) = I(X;Y,Z) + I(Y;Z).$$

Therefore, obtaining (U^N, V^N) from $\widetilde{Y}^N, \widetilde{Z}^N$ via the polar transformation, it follows that any monotone chain rule expansion on U^N, V^N should satisfy

$$\begin{aligned} R_1 &\geq I(\widetilde{X};\widetilde{Y}) = I(X;Y), \\ R_2 &\geq I(\widetilde{X};\widetilde{Z}) = I(X;Z), \\ R_1 + R_2 &= I(\widetilde{X};\widetilde{Y},\widetilde{Z}) = I(X;Y,Z) + I(Y;Z), \end{aligned}$$

with equality for the first inequality if $S^{2N} = (U^N, V^N)$ and equality for the second inequality if $S^{2N} = (V^N, U^N)$. Thus, the corner points can be achieved, and achieving any other point on the dominant face follows directly from Theorem 4. This completes the proof.

8.4 Concluding Remarks

We have presented in this chapter a polar coding scheme that achieves the entire El Gamal–Cover inner bound for the multiple description coding problem. We have seen that choosing different decoding orders can achieve different points on the dominant face of the rate region. Also, a crucial step in the proof is the independence of the two descriptions which can be ensured through a dithering argument.

Acknowledgement

Chapter 8, in part, is a reprint with permission of the material as it appears in the paper: Alankrita Bhatt, Nadim Ghaddar, and Lele Wang, "Polar coding for multiple descriptions using monotone chain rules," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 565-571, Oct 2017. The dissertation author was the primary investigator and author of this paper.

Bibliography

[1] R. J. McEliece, “Are turbo-like codes effective on nonstandard channels?” *IEEE Inf. Theory Soc. Newslett.*, vol. 51, no. 4, pp. 1–8, Dec. 2001.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1,” in *Proc. IEEE Int. Conf. on Comm.*, vol. 2, 1993, pp. 1064–1070 vol.2.

[3] R. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[4] D. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[5] D. Spielman, “Linear-time encodable and decodable error-correcting codes,” *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1723–1731, 1996.

[6] E. Arıkan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[7] S. Kudekar, T. Richardson, and R. L. Urbanke, “Spatially coupled ensembles universally achieve capacity under belief propagation,” *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, 2013.

[8] J. Zander and P. Mähönen, “Riding the data tsunami in the cloud : Myths and challenges in future wireless access,” *IEEE Comm. Magazine*, vol. 51, pp. 145–151, 03 2013.

[9] C. E. Shannon, “A mathematical theory of communication,” *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 27(4):623–656, 1948.

[10] A. Feinstein, “A new basic theorem of information theory,” *IRE Trans. Inf. Theory*, vol. 4, no. 4, pp. 2–22, 1954.

[11] R. Gallager, “A simple derivation of the coding theorem and some applications,” *IEEE Trans. Inf. Theory*, vol. 11, no. 1, pp. 3–18, 1965.

[12] P. Elias, “Coding for noisy channels,” *IRE Conv. Rec.*, vol. 3, pp. 37–46, 1955.

[13] T. Cover and J. Thomas, *Elements of Information Theory.* New York: Wiley, 2006.

[14] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*, 2nd ed. Cambridge University Press, 2011.

[15] R. W. Hamming, “Error detecting and error correcting codes,” *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, 1950.

[16] M. J. E. Golay, “Notes on digital coding,” *Proc. IRE*, vol. 37, p. 657, 6 1949.

[17] I. Reed, “A class of multiple-error-correcting codes and the decoding scheme,” *IRE Trans. Inf. Theory*, vol. 4, no. 4, pp. 38–49, 1954.

[18] D. E. Muller, “Application of boolean algebra to switching circuit design and to error detection,” *IRE Trans. Electronic Computers*, vol. EC-3, no. 3, pp. 6–12, 1954.

[19] A. Hocquenghem, “Codes correcteurs d’erreurs,” *Chiffres*, vol. 2, pp. 147–156, 1959.

[20] R. Bose and D. Ray-Chaudhuri, “On a class of error correcting binary group codes,” *Info. and Control*, vol. 3, no. 1, pp. 68–79, 1960.

[21] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *J. SIAM*, vol. 8, no. 2, pp. 300–304, 1960.

[22] M. Sipser and D. Spielman, “Expander codes,” *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1710–1722, 1996.

[23] G. Poltyrev, “Bounds on the decoding error probability of binary linear codes via their spectra,” *IEEE Trans. Inf. Theory*, vol. 40, no. 4, pp. 1284–1292, 1994.

[24] N. Shulman and M. Feder, “Random coding techniques for nonrandom codes,” *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2101–2104, 1999.

[25] S. Kudekar, S. Kumar, M. Mondelli, H. D. Pfister, E. Şaşoğlu, and R. L. Urbanke, “Reed–Muller codes achieve capacity on erasure channels,” *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4298–4316, 2017.

[26] G. Reeves and H. D. Pfister, “Reed-Muller codes achieve capacity on BMS channels,” 2021. [Online]. Available: https://arxiv.org/abs/2110.14631

[27] T. Cover, “Broadcast channels,” *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 2–14, 1972.

[28] A. El Gamal and Y.-H. Kim, *Network Information Theory.* Cambridge: Cambridge University Press, 2011.

[29] 3GPP, “NR: Multiplexing and Channel Coding,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.212, 7 2020, version 16.2.0.

[30] M. Mushkin and I. Bar-David, “Capacity and coding for the Gilbert-Elliott channels,” *IEEE Trans. Inf. Theory*, vol. 35, no. 6, pp. 1277–1290, 1989.

[31] E. Biglieri, J. Proakis, and S. Shamai, “Fading channels: information-theoretic and communications aspects,” *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2619–2692, Oct 1998.

[32] A. Lapidoth and P. Narayan, “Reliable communication under channel uncertainty,” *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2148–2177, Oct 1998.

[33] N. Ghaddar, S. Ganguly, L. Wang, and Y.-H. Kim, “A Lego-brick approach to coding for network communication,” 2022. [Online]. Available: https://arxiv.org/abs/2211.07208

[34] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[35] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec*, vol. 4, no. 142-163, p. 1, 1959.

[36] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[37] S. Gelfand and M. Pinsker, "Coding for channel with random parameters," in *Probl. Control Inf. Theory*, 1980, pp. 19–31.

[38] H. S. Witsenhausen, "On source networks with minimal breakdown degradation," *Bell Syst. Tech. J.*, vol. 59, no. 6, pp. 1083–1087, 1980.

[39] T. Berger, "Multiterminal source coding," in *The Information Theory Approach to Communications.* Springer-Verlag, New York, 1978, pp. 171–231.

[40] C. E. Shannon, "Two-way communication channels," in *Proc. 4TH Berkeley Symp. Math. Statist. Probab.*, 1961, pp. 611–644.

[41] K. Marton, "A coding theorem for the discrete memoryless broadcast channel," *IEEE Trans. Inf. Theory*, vol. 25, no. 3, pp. 306–311, 1979.

[42] O. Simeone, A. Maeder, M. Peng, O. Sahin, and W. Yu, "Cloud radio access network: Virtualizing wireless access for dense heterogeneous systems," *J. Comm. Networks.*, vol. 18, no. 2, pp. 135–149, 2016.

[43] N. Ghaddar, Y.-H. Kim, L. B. Milstein, L. Ma, and B. K. Yi, "Joint channel estimation and coding over channels with memory using polar codes," *IEEE Trans. Comm.*, vol. 69, no. 10, pp. 6575–6589, 2021.

[44] A. Bhatt, N. Ghaddar, and L. Wang, "Polar coding for multiple descriptions using monotone chain rules," in *Proc. 55th Ann. Allerton Conf. Comm. Control Comput.*, 2017, pp. 565–571.

[45] R. Gallager, *Information Theory and Reliable Communication.* New York: John Wiley and Sons, 1968.

[46] J. Chen, D.-k. He, A. Jagmohan, L. A. Lastras-Montano, and E.-h. Yang, "On the linear codebook-level duality between Slepian–Wolf coding and channel coding," *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5575–5590, 2009.

[47] C. Bennett, P. Shor, J. Smolin, and A. Thapliyal, "Entanglement-assisted capacity of a quantum channel and the reverse Shannon theorem," *IEEE Trans. Inf. Theory*, vol. 48, no. 10, pp. 2637–2655, 2002.

[48] P. Cuff, "Distributed channel synthesis," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7071–7096, 2013.

[49] S. B. Korada and R. L. Urbanke, "Polar codes are optimal for lossy source coding," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1751–1768, 2010.

[50] T. Cover and A. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inf. Theory*, vol. 25, no. 5, pp. 572–584, 1979.

[51] A. D. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inf. Theory*, vol. 20, no. 1, pp. 2–10, Jan. 1974.

[52] T. Ancheta, "Syndrome-source-coding and its universal generalization," *IEEE Trans. Inf. Theory*, vol. 22, no. 4, pp. 432–436, 1976.

[53] L. Wang and Y. Kim, "Linear code duality between channel coding and slepian-wolf coding," in *Proc. 53rd Ann. Allerton Conf. Comm. Control Comput.*, 2015, pp. 147–152.

[54] M. Mondelli, S. H. Hassani, I. Sason, and R. L. Urbanke, "Achieving marton's region for broadcast channels using polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 2, pp. 783–800, 2015.

[55] S. Ganguly, L. Wang, and Y.-H. Kim, "A functional construction of codes for multiple access and broadcast channels," in *Proc. IEEE Int. Symp. Inf. Theory*, 2020, pp. 1581–1586.

[56] N. Ghaddar, S. Ganguly, L. Wang, and Y.-H. Kim, "A Lego-brick approach to lossy source coding," in *Proc. Canadian Workshop Inf. Theory*, 2022, pp. 45–50.

[57] ——, "A Lego-brick approach to coding for asymmetric channels and channels with state," in *Proc. IEEE Int. Symp. Inf. Theory*, 2021, pp. 1367–1372.

[58] E. Arıkan, "Polar coding for the Slepian-Wolf problem based on monotone chain rules," in *Proc. IEEE Int. Symp. Inf. Theory*, 2012, pp. 566–570.

[59] S. B. Korada, "Polar codes for channel and source coding," Ph.D. Thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2009.

[60] Q. Shi, L. Song, C. Tian, J. Chen, and S. Dumitrescu, "Polar codes for multiple descriptions," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 107–119, 2015.

[61] E. Şaşoğlu, E. Telatar, and E. Yeh, "Polar codes for the two-user binary-input multiple-access channel," in *Proc. IEEE Inf. Theory Workshop*, 2010, pp. 1–5.

[62] E. Abbe and E. Telatar, "Polar codes for the m-user multiple access channel," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5437–5448, 2012.

[63] L. Wang and E. Şaşoğlu, "Polar coding for interference networks," in *Proc. IEEE Int. Symp. Inf. Theory*, 2014, pp. 311–315.

[64] R. Blasco-Serrano, R. Thobaben, M. Andersson, V. Rathi, and M. Skoglund, "Polar codes for cooperative relaying," *IEEE Trans. Comm.*, vol. 60, no. 11, pp. 3263–3273, 2012.

[65] L. Wang, "Polar coding for relay channels," in *Proc. IEEE Int. Symp. Inf. Theory*, 2015, pp. 1532–1536.

[66] Y. Matsunaga and H. Yamamoto, "A coding theorem for lossy data compression by LDPC codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2225–2229, 2003.

[67] J. Muramatsu and S. Miyake, "Hash property and coding theorems for sparse matrices and maximum-likelihood coding," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2143–2167, 2010.

[68] M. Wainwright and E. Maneva, "Lossy source encoding via message-passing and decimation over generalized codewords of LDGM codes," in *Proc. IEEE Int. Symp. Inf. Theory*, 2005, pp. 1493–1497.

[69] V. Aref, N. Macris, and M. Vuffray, "Approaching the rate-distortion limit with spatial coupling, belief propagation, and decimation," *IEEE Trans. Inf. Theory*, vol. 61, no. 7, pp. 3954–3979, 2015.

[70] S. Kumar, A. Vem, K. Narayanan, and H. D. Pfister, "Spatially-coupled codes for side-information problems," in *Proc. IEEE Int. Symp. Inf. Theory*, 2014, pp. 516–520.

[71] M. Costa, "Writing on dirty paper (corresp.)," *IEEE Trans. Inf. Theory*, vol. 29, no. 3, pp. 439–441, 1983.

[72] U. Erez, S. Shamai, and R. Zamir, "Capacity and lattice strategies for canceling known interference," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3820–3833, 2005.

[73] N. Sommer, M. Feder, and O. Shalvi, "Low-density lattice codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1561–1585, 2008.

[74] O. Shalvi, N. Sommer, and M. Feder, "Signal codes: Convolutional lattice codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5203–5226, 2011.

[75] E. Weiss, "Compression and coding (corresp.)," *IRE Trans. Inf. Theory*, vol. 8, pp. 256–257, 1962.

[76] S.-Y. Tung, "Multiterminal source coding," Ph.D. Thesis, Cornell University, Ithaca, NY, 1978.

[77] H. S. Witsenhausen and A. D. Wyner, "Source coding for multiple descriptions ii: A binary source," *Bell Syst. Tech. J.*, vol. 60, no. 10, pp. 2281–2292, 1981.

[78] J. K. Wolf, A. D. Wyner, and J. Ziv, "Source coding for multiple descriptions," *Bell Syst. Tech. J.*, vol. 59, no. 8, pp. 1417–1426, 1980.

[79] A. Gamal and T. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inf. Theory*, vol. 28, no. 6, pp. 851–857, 1982.

[80] J. Chen, C. Tian, T. Berger, and S. S. Hemami, "Multiple description quantization via Gram–Schmidt orthogonalization," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5197–5217, 2006.

[81] T. Berger and Z. Zhang, "Minimum breakdown degradation in binary source encoding," *IEEE Trans. Inf. Theory*, vol. 29, no. 6, pp. 807–814, 1983.

[82] N. Ghaddar, S. Ganguly, L. Wang, and Y.-H. Kim, "Lego-brick approach to coding for network communication," 2022. [Online]. Available: https://github.com/nadimgh/lego-brick

[83] A. Padakandla and S. S. Pradhan, "Nested linear codes achieve Marton's inner bound for general broadcast channels," in *Proc. IEEE Int. Symp. Inf. Theory*, 2011, pp. 1554–1558.

[84] R. Ahlswede, "Multiway communication channels," *Proc. 2nd Int. Symp. Inf. Theory*, pp. 23–52, 1971.

[85] D. Slepian and J. K. Wolf, "A coding theorem for multiple access channels with correlated sources," *Bell Syst. Tech. J.*, vol. 52, no. 7, pp. 1037–1076, 1973.

[86] M. Joham, W. Utschick, and J. Nossek, "Linear transmit processing in MIMO communications systems," *IEEE Trans. Sig. Processing*, vol. 53, no. 8, pp. 2700–2712, 2005.

[87] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.

[88] S. Luke, *Essentials of Metaheuristics*, 2nd ed. Lulu, 2013.

[89] J. Lee and N. Jindal, "Dirty paper coding vs. linear precoding for MIMO broadcast channels," in *Proc. 40th Asilomar Conf. Sig. Syst. Comput.*, 2006, pp. 779–783.

[90] N. Jindal and A. Goldsmith, "Dirty-paper coding versus TDMA for MIMO broadcast channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1783–1794, 2005.

[91] C. Johnson, *5G New Radio in Bullets.* Independently Published, 2019.

[92] 3GPP, "NR: Study on Integrated Access and Backhaul," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.874, 11 2018, version 0.7.0.

[93] S. Ganguly, S.-E. Hong, and Y.-H. Kim, "On the capacity regions of cloud radio access networks with limited orthogonal fronthaul," *IEEE Trans. Inf. Theory*, vol. 67, no. 5, pp. 2958–2988, 2021.

[94] P. Patil and W. Yu, "Generalized compression strategy for the downlink cloud radio access network," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6766–6780, 2019.

[95] Y. Zhou, Y. Xu, W. Yu, and J. Chen, "On the optimal fronthaul compression and decoding strategies for uplink cloud radio access networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 7402–7418, 2016.

[96] X. Yi and N. Liu, "An achievability scheme for downlink multicell processing with finite backhaul capacity: The general case," in *IEEE Int. Conf. on Wireless Comm. and Sig. Processing*, 2015, pp. 1–5.

[97] S.-H. Park, O. Simeone, O. Sahin, and S. Shamai Shitz, "Fronthaul compression for cloud radio access networks: Signal processing advances inspired by network information theory," *IEEE Sig. Processing Magazine*, vol. 31, no. 6, pp. 69–79, 2014.

[98] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* Cambridge, MA, USA: MIT Press, 1992.

[99] Y. Zhou and W. Yu, "Optimized backhaul compression for uplink cloud radio access network," *IEEE Journ. Sel. Are. Comm.*, vol. 32, no. 6, pp. 1295–1307, 2014.

[100] A. Sanderovich, O. Somekh, H. V. Poor, and S. Shamai, "Uplink macro diversity of limited backhaul cellular network," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3457–3478, 2009.

[101] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.

[102] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[103] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inf. Theory*, vol. 6, no. 1, pp. 7–12, 1960.

[104] T. Berger and J. Gibson, "Lossy source coding," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2693–2723, 1998.

[105] 3GPP, "Physical Layer Standard for cdma2000 Spread Spectrum Systems," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 3GPP2 C.S0002-A, 2 2002, version 6.0.

[106] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, 1960.

[107] E. O. Elliott, “Estimates of error rates for codes on burst-noise channels,” *Bell Syst. Tech. J.*, vol. 42, no. 5, pp. 1977–1997, 1963.

[108] P. Sadeghi, R. A. Kennedy, P. B. Rapajic, and R. Shams, “Finite-state markov modeling of fading channels - a survey of principles and applications,” *IEEE Sig. Processing Magazine*, vol. 25, no. 5, pp. 57–80, 2008.

[109] I. Abou-Faycal, M. Medard, and U. Madhow, “Binary adaptive coded pilot symbol assisted modulation over Rayleigh fading channels without feedback,” *IEEE Trans. Comm.*, vol. 53, no. 6, pp. 1036–1046, 2005.

[110] S. Akin and M. C. Gursoy, “Training optimization for Gauss-Markov rayleigh fading channels,” in *Proc. IEEE Int. Conf. on Comm.*, 2007, pp. 5999–6004.

[111] R. H. Clarke, “A statistical theory of mobile-radio reception,” *Bell Syst. Tech. J.*, vol. 47, no. 6, pp. 957–1000, 1968.

[112] R. G. Gallager, *Principles of Digital Communication.* Cambridge University Press, 2008.

[113] I. Tal and A. Vardy, “List decoding of polar codes,” *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[114] E. Şaşoğlu, “Polarization in the presence of memory,” in *Proc. IEEE Int. Symp. Inf. Theory*, July 2011, pp. 189–193.

[115] ——, “Polar coding theorems for discrete systems,” Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, 2011.

[116] B. Shuval and I. Tal, “Fast polarization for processes with memory,” *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2004–2020, April 2019.

[117] R. Wang, R. Liu, and Y. Hou, “Joint successive cancellation decoding of polar codes over intersymbol interference channels,” *CoRR*, vol. abs/1404.3001, 2014. [Online]. Available: http://arxiv.org/abs/1404.3001

[118] R. Wang, J. Honda, H. Yamamoto, R. Liu, and Y. Hou, “Construction of polar codes for channels with memory,” in *Proc. IEEE Inf. Theory Workshop*, Oct 2015, pp. 187–191.

[119] L. Wang, “Channel coding techniques for network communication,” Ph.D. Thesis, University of California, San Diego, La Jolla, CA, 2015.

[120] 3GPP, “5G; New Radio; Physical channels and modulation,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.211, 07 2018, version 15.2.0.

[121] D. Manolakis, V. Ingle, and S. Kogon, *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering, and Array Processing*, ser. Artech House signal processing library. Artech House, 2005.

[122] K. E. Baddour and N. C. Beaulieu, “Autoregressive modeling for fading channel simulation,” *IEEE Wireless Comm.*, vol. 4, no. 4, pp. 1650–1662, 2005.

[123] V. Bioglio, F. Gabry, and I. Land, “Low-complexity puncturing and shortening of polar codes,” in *Proc. IEEE Wireless Comm. and Networking Conf. Workshop*, 2017, pp. 1–6.

[124] T. Wang, D. Qu, and T. Jiang, “Parity-check-concatenated polar codes,” *IEEE Comm. Letters*, vol. 20, no. 12, pp. 2342–2345, 2016.

[125] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Trans. Comm.*, vol. 60, no. 11, pp. 3221–3227, 2012.

[126] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[127] Meng-Han Hsieh and Che-Ho Wei, "Channel estimation for OFDM systems based on comb-type pilot arrangement in frequency selective fading channels," *IEEE Trans. Consumer Electronics*, vol. 44, no. 1, pp. 217–225, 1998.

[128] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Trans. Broadcasting*, vol. 48, no. 3, pp. 223–229, 2002.

[129] N. Ghaddar, Y. Kim, L. B. Milstein, L. Ma, and B. K. Yi, "Joint channel estimation and error correction for finite-state Markov channels using polar codes," in *IEEE Global Comm. Conf.*, 2018, pp. 1–6.

[130] E. Şaşoğlu, E. Telatar, and E. Yeh, "Polar codes for the two-user multiple-access channel," in *IEEE Trans. Inf. Theory*, vol. 59, no. 10, October 2013, pp. 6583–6592.

[131] E. Arıkan, "Polar coding for the Slepian-Wolf problem based on monotone chain rules," in *Proc. IEEE Int. Symp. Inf. Theory*, July 2012, pp. 566–570.

[132] Q. Shi, L. Song, C. Tian, J. Chen, and S. Dumitrescu, "Polar codes for multiple descriptions," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 107–119, January 2015.

[133] J. Wang, J. Chen, L. Zhao, P. Cuff, and H. Permuter, "On the role of the refinement layer in multiple description coding and scalable coding," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1443–1456, March 2011.