DISSERTATION

BLIND PERFORMANCE ESTIMATION AND QUANTIZER DESIGN WITH APPLICATIONS TO RELAY NETWORKS

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften

unter der Leitung von
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Gerald Matz
Institute of Telecommunications

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von
Andreas Winkelbauer
Landstraßer Gürtel 21/14
1030 Wien

Wien, im Dezember 2014 _____

Die Begutachtung dieser Arbeit erfolgte durch:

1. Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Gerald Matz

   Institute of Telecommunications
   Technische Universität Wien

2. Prof. Dr. sc. techn. Andreas Burg

   Telecommunications Circuits Laboratory
   École polytechnique fédérale de Lausanne

To Veronika♡

Abstract

In this thesis, we introduce blind estimators for several performance metrics of Bayesian detectors, we study rate-information-optimal quantization and introduce algorithms for quantizer design in the communications context, and we apply our results to a relay-based cooperative communication scheme.

After a discussion of the background material which serves as a basis for this thesis, we study blind performance estimation for Bayesian detectors. We consider simple binary and M-ary hypothesis tests and introduce blind estimators for the conditional and unconditional error probabilities, the minimum mean-square error (MSE), and the mutual information. The proposed blind estimators are shown to be unbiased and consistent. Furthermore, we compare the blind estimators for the error probabilities to the corresponding nonblind estimators and we give conditions under which the blind estimators dominate their respective nonblind counterpart for arbitrary distributions of the data. In particular, we show that the blind estimator for the unconditional error probability always dominates the corresponding nonblind estimator in terms of the MSE. Subsequently, the Cramér-Rao lower bound for bit error probability estimation under maximum *a posteriori* detection is derived. Moreover, it is shown that an efficient estimator does not exist for this problem. Application examples conclude the discussion of blind performance estimators.

We then introduce an approach to quantization that we call rate-information quantization. The main idea of rate-information-optimal quantization is to compress data such that its quantized representation is as informative as possible about another random variable. This random variable is called the relevance variable and it is correlated with the data. The rate-information approach is well suited for communication problems, which is in contrast to rate-distortion (RD) quantization. We focus on the case where the data and the relevance variable are jointly Gaussian and we derive closed-form expressions for the optimal trade-off between the compression rate and the preserved information about the relevance variable. It is then shown that the optimal rate-information trade-off is achieved by suitable linear preprocessing of the data with subsequent MSE-optimal source coding. This result connects RD theory, the Gaussian information bottleneck, and minimum MSE estimation. Furthermore, we show

that the asymptotic rate-information trade-off can be closely approached using optimized scalar quantizers.

Next, we consider quantization in a communications context and we introduce algorithms which allow us to design quantizers that maximize the achievable rate. One of our algorithms operates in a similar manner as the famous Lloyd-Max algorithm, but it maximizes the mutual information instead of minimizing the average distortion. Moreover, we propose a greedy algorithm for scalar quantizer design which is conceptually simple and computationally attractive. Subsequently, the concept of channel-optimized vector quantization, which is a well-known approach to joint source-channel coding, is extended to mutual information as optimality criterion. The resulting optimization problem is solved using an algorithm that is based on the information bottleneck method. To conclude the discussion of quantization for communication problems, we compare the proposed algorithms and provide application examples.

Finally, we apply our results to a cooperative transmission scheme for the multiple-access relay channel with two or more sources. In this scheme, the relay quantizes the received signals and performs network encoding of the quantized data. The quantizers at the relay are optimized using our Lloyd-Max-like algorithm. The network encoder is designed using a suitably modified version of the previously introduced algorithm for channel-optimized vector quantizer design. The relay operations are simple to implement and allow the considered transmission scheme to scale well with the number of sources. We provide numerical results that confirm the excellent performance of our scheme and underpin the usefulness of the proposed blind performance estimators.

Kurzfassung

In dieser Dissertation werden blinde Schätzer für einige Gütekriterien von Bayesschen Detektoren beschrieben, es wird rate-information-optimale Quantisierung diskutiert, und es werden Algorithmen zum Entwurf von Quantisierern im Bereich der Datenübertragung eingeführt. Die in diesen Bereichen erzielten Ergebnisse werden schließlich auf ein Relais-basiertes kooperatives Übertragungsverfahren angewandt.

Nach einer Abhandlung der für diese Dissertation notwendigen Grundlagen behandeln wir blinde Güteschätzung für Bayessche Detektoren. Wir betrachten binäre sowie M-fache Hypothesentests und stellen blinde Schätzer für die bedingten und unbedingten Fehlerwahrscheinlichkeiten, für den minimalen mittleren quadratischen Fehler und für die Transinformation vor. Es wird gezeigt, dass die vorgeschlagenen blinden Schätzer erwartungstreu und konsistent sind. Weiters vergleichen wir die blinden Schätzer für die Fehlerwahrscheinlichkeiten mit den entsprechenden nicht-blinden Schätzern. Wir geben Bedingungen an, unter denen die blinden Schätzer ihre nicht-blinden Pendants für beliebige Verteilungen der Daten dominieren. Insbesondere zeigen wir, dass der blinde Schätzer für die unbedingte Fehlerwahrscheinlichkeit den entsprechenden nicht-blinden Schätzer bezüglich des mittleren quadratischen Fehlers stets dominiert. Anschließend wird die Cramér-Rao-Schranke für die Schätzung der Bitfehlerwahrscheinlichkeit eines Maximum-a-posteriori-Detektors hergeleitet. Darüber hinaus wird gezeigt, dass für dieses Schätzproblem kein effizienter Schätzer existiert. Wir beschließen die Untersuchung von blinden Güteschätzern mit der Diskussion einiger Anwendungsbeispiele.

Danach beschreiben wir einen Ansatz zur Quantisierung, den wir Rate-information-Quantisierung nennen. Der Grundgedanke von rate-information-optimaler Quantisierung ist, Daten derart zu komprimieren, dass die quantisierte Darstellung möglichst aussagekräftig über eine andere Zufallsvariable ist. Diese mit den Daten korrelierte Zufallsvariable bezeichnen wir als Relevanzvariable. Der Rate-information-Ansatz ist im Gegensatz zum Rate-distortion-Ansatz für Anwendungen im Bereich der Übertragungstechnik sehr gut geeignet. Wir richten unser Hauptaugenmerk auf den Fall einer Gaußschen Verbundverteilung von Daten und Relevanzvariable. In diesem Fall finden wir geschlossene Ausdrücke für den optimalen Abtausch zwischen der Quantisierungsrate und der erhalten bleibenden Information über die Relevanzvariable. Wir nennen diesen Abtausch den Rate-information-Trade-off, und wir beweisen,

dass der optimale Rate-information-Trade-off durch geeignete lineare Filterung mit anschließender, im Sinne des mittleren quadratischen Fehlers optimaler, Quellencodierung erreicht werden kann. Dieses Ergebnis stellt eine Verbindung zwischen Rate-distortion-Theorie, Gaußschem Information-bottleneck und Wiener-Filterung her. Weiters zeigen wir, dass optimierte skalare Quantisierer den asymptotischen Rate-information-Trade-off beinahe erreichen.

Im Anschluss betrachten wir Quantisierung im Kontext der Übertragungstechnik und stellen Algorithmen für den Entwurf von Quantisierern vor, welche die erreichbare Datenrate maximieren. Einer der vorgestellten Algorithmen funktioniert ähnlich wie der berühmte Lloyd-Max-Algorithmus, mit dem Unterschied, dass er nicht die Signalverzerrung minimiert sondern die Transinformation maximiert. Zudem schlagen wir einen Greedy-Algorithmus für den Entwurf skalarer Quantisierer vor, der konzeptionell einfach ist und geringen Rechenaufwand aufweist. Anschließend erweitern wir das Konzept der kanaloptimierten Vektorquantisierung, welches ein bekanntes Verfahren zur gemeinsamen Kanal- und Quellencodierung darstellt, indem wir die Transinformation als Optimalitätskriterium verwenden. Das zugehörige Optimierungsproblem wird mit Hilfe eines Algorithmus gelöst, der auf der information-bottleneck-Methode basiert. Zum Abschluss dieses Teils unserer Arbeit vergleichen wir die vorgeschlagenen Algorithmen und diskutieren Anwendungsbeispiele.

Abschließend wenden wir die zuvor gewonnenen Ergebnisse auf ein kooperatives Übertragungsverfahren für den Mehrfachzugriff-Relais-Kanal mit mindestens zwei Quellen an. In diesem Übertragungsverfahren quantisiert das Relais die Empfangssignale und wendet anschließend Netzcodierung auf die quantisierten Daten an. Für die Optimierung der Quantisierer am Relais verwenden wir unseren Lloyd-Max-artigen Algorithmus. Der Entwurf der Netzcodierung erfolgt mittels einer geeigneten Modifikation des zuvor vorgestellten Algorithmus für den Entwurf kanaloptimierter Vektorquantisierer. Die Signalverarbeitung am Relais ist einfach zu implementieren und gewährleistet, dass das betrachtete Übertragungsverfahren vorteilhaft mit der Anzahl der Quellen skaliert. Wir geben numerische Ergebnisse an, welche die hervorragende Leistungsfähigkeit unseres Übertragungsverfahrens bestätigen. Zudem untermauern diese Ergebnisse die Nützlichkeit der von uns vorgeschlagenen blinden Güteschätzer.

Acknowledgements

I have been very much looking forward to writing these lines for quite some time. A four-and-a-half-year journey is coming to an end and I could not be happier with the outcome of my PhD-endeavor. This is due to the exceptional and wonderful support I have received, which made my time as a doctoral student much more successful and also very pleasant. It's about time to say "thank you"!

First and foremost, I want to express my gratitude to Gerald. My first encounter with him was on October 4, 2007, when I was a student in his class on statistical signal processing. I quickly realized that Gerald is an extremely gifted lecturer. It was a great pleasure to observe how well he explained very complicated things in a very clear and simple manner. Unsurprisingly, his office door was the first address when I was looking for an interesting and challenging master thesis topic. At that time, I did not think about doing a PhD. However, Gerald's commitment to high-quality research and the unique spirit in his group made me rethink my plans. Fast forward to today, I can say that I absolutely don't regret my choice. Speaking in statistical decision theory terms: the observations I made have enforced the prior belief. Gerald, I am extremely grateful for your continuous support, for going above and beyond the call of duty when it comes to revising our papers (and this thesis!) late at night, for trusting in me and my capabilities, and for giving me the opportunity to pursue a PhD. Thank you very much!

I am indebted to Andreas Burg, who kindly agreed to act as referee and examiner. Thank you for your interest in my work and for your patience. It is a pleasure working with you and I would be glad to intensify our collaboration in the future.

I am grateful to my colleagues at the Institute of Telecommunications. Our institute is a very vibrant place, which makes everyday work just so much more pleasant. The flat hierarchy and the open door culture at our institute are most appreciated. I'd like to thank all our professors for their great teaching. Special thanks go to Franz Hlawatsch and Norbert Görtz. Franz, thank you for teaching me all the basic things that I now use on a daily basis. If there is an ideal teacher, then it's most probably you. Norbert, thank you for the collaboration and for teaching me everything I know about source coding.

I am thankful to all of my past and present colleagues in the CT group for providing an outstanding working environment: Peter, Joe, Clemens, Günter, Stefan S., FloX, Valentin, Mohsen, Maxime, Georg, Michael, and Stefan F. (in pseudo-chronological order). It was a great pleasure to get to know Vaughan Clarkson, who was brave enough to spend his sabbatical with the CT group. Vaughan, you are a great guy and you made sure that I will never forget the 2013 Asilomar conference. Thank you!

I want to thank the Mensa crew (Gregor and Günther being the core team nowadays) for the many remarkable meals we had together. In the presence of you guys, even the weirdest and saltiest creations of the Mensa chefs become digestible.

One of my favorite quotes, which is attributed to Confucius, is "choose a job you love, and you will never have to work a day in your life." I am lucky, since I indeed love my job. However, life is not just about work and I feel very blessed that I am surrounded by a lovely family and wonderful friends in my private life.

Meinen Eltern, Barbara und Helmut, bin ich zutiefst dankbar für Ihre ständige und bedingungslose Unterstützung. Ihr habt mir die Freiheit gegeben jenen Dingen nachzugehen die mir wichtig sind. Danke, dass Ihr immer an mich glaubt und jederzeit für mich da seid.

Я хочу выразить слова благодарности Наталье и Олегу за то, что они сердечно приняли меня в свою семью, за гостеприимство и неустанный интерес к прогрессу моей работы. Поездка по Узбекистану произвела на меня очень яркое впечатление и я с нетерпением жду повторения наших незабываемых путешествий.

More than to anyone else I owe my deepest gratitude to my fiancée Veronika. Over six years ago, we started a wonderful journey, the best journey of my life, and I am very much looking forward to the great experiences that are ahead of us. Thanks to Veronika's unconditional support, endless patience, and genuine love I can now write these acknowledgements. There are no words to describe how much I am indebted to her. Большое спасибо. Я очень тебя люблю!

In case I forgot to mention *you* above: I am terribly sorry! In my defense, let me remark that it is past 6 a.m. (no, I am not an early bird!) and I am doing my best to write something that makes at least a little sense.

Vienna, December 4, 2014.
Andreas Winkelbauer

Contents

1

Introduction

Communication systems and especially wireless communication technologies have become an integral part of our everyday life. Today, many people around the world are so accustomed to being always "online" that a few days without internet connection is like a nightmare for them. However, according to estimates of the International Telecommunication Union, only 40 % of the global population will have internet access by the end of 2014. Therefore, the importance and the size of wireless communication networks will continue to grow at a fast pace for the foreseeable future. This is especially true for developing countries where for every user who is online, there are two users who are not. Connecting the next 4 billion people to the internet will give rise to tremendous technological and societal challenges in the next decade. Furthermore, billions of interconnected sensors and embedded computing devices, forming the *internet of things* (IoT), are expected to be used in applications such as intelligent transportation systems, home automation, and energy management. The flood of data produced by the IoT will pose substantial challenges for future data compression techniques and data storage systems.

The developments mentioned above provide untold research opportunities in the areas of communication and information theory, and signal processing. Current research themes aiming to improve wireless networks in terms of spectral efficiency and energy efficiency include cooperative communications, interference management, large-scale antenna systems, cognitive radio, and resource management with cross-layer optimization. These topics are not purely of academic interest; cooperative transmission techniques have for example found their way into communication standards. The IEEE 802.16j standard [45] introduced multihop relaying for WiMAX, and IEEE 802.11ah [44] will (presumably in early 2016) introduce relay access points for WiFi networks operating in unlicensed sub-gigahertz frequency bands. Another example is the 3GPP Release 10 (also known as LTE-A) which introduces coordinated multipoint transmission and relay nodes [1]. While future networks will have to

go considerably beyond traditional point-to-point communication paradigms to satisfy consumer demands, there are still a lot of relevant open problems concerning the physical layer of point-to-point links. For example the information-theoretic limits and the optimal design of receiver front-ends (consisting of everything from the antenna to the analog-to-digital converter) have not been sufficiently studied to date. However, carefully optimizing, e.g., synchronization, sampling, and analog-to-digital conversion in wireless receivers may yield substantial performance and energy improvements over the current state of the art.

1.1 Motivation and Problem Statement

In this thesis we study blind performance estimation, data compression and quantizer design, and relay-based cooperative communication. We next give a brief motivation for our work on each of these topics.

Blind Performance Estimation for Bayesian Detectors. An exact analytical performance analysis of Bayesian hypothesis tests is often difficult or even impossible. Alternatively, performance bounds can be considered which are usually easier to derive and to evaluate. If bounds do not provide sufficient insight and an exact analysis remains elusive, Monte Carlo simulations provide a way to estimate the performance of detectors. When soft information is used at the detector, performance estimation can be carried out in a blind manner [43]. Here, "blind" means that the estimator may only use the data that is available to the hypothesis test, i.e., it does not have access to the true hypothesis or any other side information. Therefore, blind estimators are not restricted to simulations; they are suitable for online performance analysis of soft-information-based detectors. This is important, e.g., for receivers in communication systems which employ adaptive modulation and coding techniques. Our main goal is to find and analyze blind estimators for several performance metrics of simple Bayesian hypothesis tests.

The Rate-Information Trade-off in the Gaussian Case. In digital communication systems, the receiver has to employ some form of quantization, e.g., analog-to-digital conversion. Clearly, there is a trade-off between the quantizer resolution and the amount of information that can reliably be decoded after quantization. Closed-form expressions for this trade-off (termed *rate-information trade-off*) are available only in a few special cases [112]. The information bottleneck (IB) method [97] allows us to numerically compute the rate-information trade-off for discrete random variables. In this thesis, we study the rate-information trade-off for Gaussian channels with Gaussian input. Our main goals are to derive closed-form expressions for the rate-information trade-off and to find connections to rate-distortion (RD) theory [8]. Although motivated by a communication problem, our work applies to the compression of arbitrary jointly Gaussian data sets.

Quantizer Design for Communication Problems. Quantizer design is well studied in the lossy source coding setting. RD theory provides fundamental performance limits and there exist algorithms, e.g., the Lloyd-Max algorithm [66, 71] and the LBG algorithm [65], which allow us to find optimized quantizers. However, a lossy source coding perspective is generally not appropriate in a communications context where we are interested in maximizing the data rate rather than in representing a signal with small distortion. Hence, we aim for quantizers which are optimized in the sense of maximum mutual information, i.e., we are interested in rate-information quantization instead of RD quantization. While the IB method allows us to numerically compute the fundamental limits for rate-information quantization, it does not yield deterministic quantizers for finite blocklengths. We note that mutual-information-based quantizer design is substantially different from distortion-based quantizer design. Our main goal is to conceive algorithms for the design of low-rate quantizers for communication problems.

Quantization-Based Network Coding for the Multiple-Access Relay Channel (MARC). Cooperative communication strategies have been recognized as a promising way to improve spectral efficiency and to increase reliability in communication networks. When direct user cooperation [92,93] is not feasible, relay nodes can be employed whose sole purpose is to facilitate the users' transmissions. For example in cellular networks, relays can provide diversity and improve coverage for cell-edge users [114]. In networks with more than one data stream, network coding [2] enables increased throughput by coding data at intermediate nodes instead of simple forwarding. Well-known relaying protocols include amplify-and-forward (AF), decode-and-forward (DF), and compress-and-forward [19]. The AF scheme is easy to implement, but has the drawback of analog transmission which requires highly linear and thus inefficient power amplifiers at the relay. DF avoids this drawback and can easily be combined with network coding, but it has increased complexity and delay due to decoding at the relay. Our main goal is to devise a compression-based transmission scheme for the MARC with two or more users which is simple to implement and incorporates network coding at the physical layer.

1.2 Organization of this Thesis

In the remainder of this chapter, we summarize our major original contributions and we present the notation we use in this thesis.

Chapter 2 covers the background material which serves as a basis for the subsequent chapters. The material in this chapter is presented in such a way that the reader can quickly recall the most important definitions and results without having to browse through the literature.

In **Chapter 3** we first give an example for blind estimation of the bit error probability. We next consider the binary case and study the properties of log-likelihood ratios (LLRs). We

then formulate simple blind estimators for a number of performance metrics. Next, we analyze the mean-square error (MSE) performance of the proposed estimators and compare them to corresponding nonblind estimators. Moreover, for the case of conditionally Gaussian LLRs, we derive the Cramér-Rao lower bound (CRLB) [21, 84] for bit error probability estimation. Finally, we provide application examples to corroborate the usefulness of the proposed blind estimators. Parts of the material in this chapter have been published in [107].

Chapter 4 studies the rate-information trade-off for jointly Gaussian random vectors. After formalizing the rate-information trade-off, we review the Gaussian information bottleneck (GIB) [17]. Using the GIB, we find closed-form expressions for the rate-information trade-off in the scalar case and in the vector case. Next, we study connections between the rate-information trade-off and RD theory and we prove that the GIB can be decomposed into linear filtering with subsequent MSE-optimal quantization. Finally, we design quantizers and compare their performance to the asymptotic limit. The material in this chapter has in part been published in [72, 104, 110].

In **Chapter 5** we consider the design of optimized quantizers in the sense of maximum mutual information. First, we point out the differences between this problem and distortion-based quantization, and we review the Lloyd-Max algorithm [66, 71]. Next, we conceive an algorithm which is strongly reminiscent of the famous Lloyd-Max algorithm but maximizes mutual information instead of minimizing the MSE. Furthermore, we discuss the design of scalar quantizers using a greedy algorithm which are simple to implement. We present an algorithm for channel-optimized vector quantization (COVQ) which is based on the IB method. This algorithm includes the design of scalar quantizers and vector quantizers as special cases. Finally, we give application examples and provide a comparison of the proposed algorithms. Parts of the material in this chapter have been published in [111].

Chapter 6 presents a physical layer network coding scheme for the MARC. After an introductory discussion of related work, we present the system model and explain the basic operation of all nodes. Next, we describe the coding strategy at the relay in detail. The relay essentially performs LLR quantization followed by a network encoding operation. The network encoder design is based on a suitably modified version of the COVQ algorithm introduced in Chapter 5. We then present an iterative message passing decoder which jointly decodes all source data at the destination. Finally, we provide simulation results which demonstrate the effectiveness of the considered transmission scheme. We also consider performance evaluation using the blind estimators introduced in Chapter 3. The material in this chapter has in part been published in [109].

Conclusions are provided in **Chapter 7**. We summarize our main findings and discuss the insights gained in this thesis. Finally, we point out several open problems which may serve as a basis for further research.

Throughout this thesis, lengthy proofs are relegated to the appendices for better readability. Appendix D provides a collection of formulas for the (raw and central) moments and absolute moments of the Gaussian distribution [103].

1.3 Original Contributions

In the following we summarize the major original contributions of this thesis.

Blind Performance Estimation for Bayesian Detectors

- We propose unbiased and consistent blind estimators for several performance metrics of Bayesian detectors. In particular, we consider (conditional) error probabilities, the minimum MSE, and mutual information. We note that our contributions go substantially beyond previous work in [43, 57, 58, 67, 96].
- We analyze and suitably bound the MSE of our blind estimators. For the (conditional) error probabilities we include a comparison to nonblind estimators. For the unconditional error probability, we prove that the blind estimator always dominates the corresponding nonblind estimator. For the conditional error probabilities, we give conditions under which the blind estimators dominate the corresponding nonblind estimators.
- We derive the CRLB for bit error probability estimation in the case of conditionally Gaussian LLRs. Furthermore, we show that in this case an efficient estimator does not exist. We numerically evaluate the MSE of our proposed bit error probability estimator and compare it to the CRLB.
- We study the properties of LLRs. In particular, we find novel relations between conditional and unconditional moments of functions of LLRs. These results prove to be useful in the derivation of the proposed blind estimators.
- We give application examples for the proposed blind estimators and confirm their usefulness using numerical simulation results. We consider suboptimal detectors and model uncertainty (e.g., imperfect channel state information in the communications context) and we show that our blind estimators produce useful results in these cases.

The Rate-Information Trade-off in the Gaussian Case

- We derive closed-form expressions for the rate-information trade-off. In particular, we characterize the information-rate function and the rate-information function, and we study the properties of these two functions. We show that in the asymptotic limit, rate-information-optimal quantization can be modeled by additive Gaussian noise.
- We prove that MSE-optimal (noisy) source coding is suboptimal in terms of the rate-information trade-off. However, we show that suitable linear preprocessing with subsequent MSE-optimal quantization is sufficient to achieve the optimal rate-information trade-off.
- Our results show that the GIB can be decomposed into linear filtering with subsequent MSE-optimal source coding. This is important because it relates the lesser-known

GIB to two much more widely known concepts. Moreover, the RD theorem provides achievability and converse for the rate-information trade-off.

- We design quantizers for fixed blocklength, and we numerically evaluate their performance and compare it to the optimal rate-information trade-off. It turns out that it is sufficient to consider MSE-optimal quantizers and the information-rate function can be closely approached with increasing quantization rate.

Quantizer Design for Communication Problems

- We derive a novel algorithm for the design of scalar quantizers that maximize mutual information. The proposed algorithm performs alternating maximization and allows for an elegant formulation in the binary case in terms of LLRs. Our algorithm is simple to implement and can directly quantize continuous random variables which is in contrast to IB-based algorithms.

- We present a simple yet effective greedy algorithm for the design of mutual-information-optimal scalar quantizers. This algorithm is attractive because it finds a locally optimal quantizer with little computational effort. In particular, the proposed greedy algorithm avoids line search and root-finding methods.

- We propose an algorithm which finds channel-optimized vector quantizers maximizing mutual information. This is the first algorithm extending the concept of COVQ to mutual information as optimality criterion. An important advantage of the proposed algorithm is that it additionally yields optimized labels for the quantizer output.

- We provide a comparison of the proposed quantizer design algorithms. In particular, we give guidelines for the selection of the appropriate algorithm and we study the convergence behavior of the proposed algorithms. Moreover, we design mutual-information-optimal quantizers and compare their performance to the optimal rate-information trade-off which is computed numerically using the IB method.

Quantization-Based Network Coding for the MARC

- We propose a physical layer network coding scheme for the MARC which supports two or more sources and is simple to implement. In our scheme, the relay performs quantization-based network encoding which essentially consists of LLR quantization and a simple table lookup operation.

- We design the network encoder at the relay as a channel-optimized vector quantizer which is optimized using a suitably modified version of the COVQ algorithm introduced in Chapter 5. This approach allows us to effectively combat the noise on the relay-destination channel and is superior to non-channel-optimized designs.

- We derive an iterative receiver which jointly decodes the network code and the channel codes using the sum-product algorithm [54]. We use a factor graph approach to describe the overall network-channel code and to derive the iterative joint network-channel decoder.

- We provide numerical results and analyze the performance of the proposed scheme using practical channel codes. We evaluate the bit and block error rates of our scheme, and compare its performance to baseline schemes. It is observed that the proposed scheme yields diversity and coding gains, and scales well beyond two sources.

1.4 Notation

We use boldface lowercase letters for column vectors and boldface uppercase letters for matrices. For random variables, we use upright sans-serif letters. Sets are denoted by calligraphic letters. The indicator function $\mathbb{1}\{\cdot\}$ equals 1 if its argument is a true statement and it equals 0 otherwise. Markov chains are denoted as $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z}$ which implies that $p(x, z|y) = p(x|y)p(z|y)$. A multivariate Gaussian (normal) distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{C}$ is denoted by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{C})$. Similarly, $\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{C})$ denotes a complex Gaussian distribution that is circularly symmetric about its mean $\boldsymbol{\mu}$ and has covariance matrix $\boldsymbol{C}$. Additional frequently used notation is summarized below.

$\mathbb{N}$	natural numbers
$\mathbb{N}_0$	nonnegative integers
$\mathbb{Z}$	integers
$\mathbb{R}$	real numbers
$\mathbb{R}_+$	nonnegative real numbers
$\mathbb{C}$	complex numbers
$\lvert\mathcal{A}\rvert$	cardinality of the set $\mathcal{A}$
z^*	complex conjugate of z
$\Re(z)$	real part of z
$[x]^+$	shorthand notation for $\max\{0, x\}$
$\log^+ x$	shorthand notation for $\log(\max\{1, x\})$
$Q(x)$	Gaussian Q-function, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-t^2/2)\mathrm{d}t$
$s(x)$	unit step function
$\delta(x)$	Dirac delta function
$\delta_{i,j}$	Kronecker delta
$\mathbf{1}$	all-ones vector
$\mathbf{0}$	all-zeros vector
$\mathrm{vec}\{\boldsymbol{A}\}$	vector consisting of the stacked columns of the matrix $\boldsymbol{A}$
$\lVert\cdot\rVert_2$	Euclidean norm

$\boldsymbol{A}^{\mathrm{T}}$	transpose of the matrix $\boldsymbol{A}$
$\boldsymbol{I}$	identity matrix
$\mathrm{diag}\{a_n\}_{n=1}^{N}$	$N \times N$ diagonal matrix with diagonal entries $a_1, \ldots, a_N$
$\mathbb{P}\{\cdot\}$	probability
$\mathbb{E}\{\cdot\}$	expectation operator
$\mathrm{var}\{\cdot\}$	variance
$D(\cdot\|\cdot)$	relative entropy
$H(\cdot)$	entropy
$h(\cdot)$	differential entropy
$h_2(p)$	binary entropy function, $h_2(p) = -p\log_2 p - (1-p)\log_2(1-p)$
$I(\cdot;\cdot)$	mutual information
$\succeq$	element-wise inequality
$\odot$	element-wise multiplication
$\oplus$	modulo-2 addition
$\boxplus$	boxplus operator (cf. [40])

2

Preliminaries

We cover a diverse set of the topics in this thesis. In this chapter, we therefore present the background material which serves as a basis for Chapters 3 to 6. We first introduce basic concepts from convex optimization (Section 2.1) and information theory (Section 2.2). Next, we consider classical parameter estimation in Section 2.3 and Bayesian hypothesis testing in Section 2.4. We then discuss factor graphs and the sum-product algorithm (Section 2.5) followed by an introduction to soft information processing and codes on graphs (Section 2.6). Finally, we review the information bottleneck (IB) method and discuss the iterative IB algorithm in Section 2.7. The main purpose of this chapter is to summarize the well-known definitions and results we use in this thesis. Thus, we state all results without proofs and refer to the literature for more details.

2.1 Convex Optimization

The material in this section is taken mostly from Boyd and Vandenberghe [12]. We begin by discussing convex sets. A set $\mathcal{S}$ is *convex* if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}$ and any $\theta \in [0,1]$, we have

$$\theta \boldsymbol{x}_1 + (1-\theta)\boldsymbol{x}_2 \in \mathcal{S}. \tag{2.1}$$

The statement in (2.1) says that the line segment between any two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ must lie in $\mathcal{S}$. Important examples of convex sets are hyperplanes, halfspaces, polyhedra, and Euclidean balls. A set of the form

$$\{\boldsymbol{x} \,|\, \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} = b\} \tag{2.2}$$

with $\boldsymbol{a} \in \mathbb{R}^n$, $\boldsymbol{a} \neq \boldsymbol{0}$, and $b \in \mathbb{R}$ is called a *hyperplane*. The hyperplane in (2.2) divides $\mathbb{R}^n$ into two halfspaces. A closed *halfspace* is a convex set which can be written as

$$\{\boldsymbol{x} \,|\, \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \leq b\}, \tag{2.3}$$

where $\boldsymbol{a} \neq \boldsymbol{0}$. An intersection of a finite number of halfspaces and hyperplanes is called a *polyhedron*. More formally, a polyhedron is a convex set of the form

$$\mathcal{P} = \{\boldsymbol{x} \,|\, \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i,\, i = 1, \ldots, m,\, \boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{x} = d_j,\, j = 1, \ldots, p\}. \tag{2.4}$$

The polyhedron in (2.4) is the intersection of m halfspaces and p hyperplanes. An important special case of a polyhedron in $\mathbb{R}^n$ is obtained by letting $\boldsymbol{a}_i = -\boldsymbol{e}_i, b_i = 0,\ i = 1, \ldots, n$, $\boldsymbol{c}_1 = \boldsymbol{1}_n$, and $d_1 = 1$ in (2.4), where $\boldsymbol{e}_i$ denotes the ith unit vector and $\boldsymbol{1}_n$ is the length-n all-ones vector. This set is an $(n-1)$-dimensional *probability simplex* which can be compactly written as

$$\mathcal{P}_{\mathcal{S}} = \{\boldsymbol{p} \,|\, \boldsymbol{p} \succeq \boldsymbol{0},\, \boldsymbol{1}_n^{\mathrm{T}}\boldsymbol{p} = 1\}. \tag{2.5}$$

In (2.5), "$\succeq$" denotes element-wise inequality. An element $\boldsymbol{p} \in \mathcal{P}_{\mathcal{S}}$ corresponds to a probability distribution on n elements. A *Euclidean ball* is a convex set of the form

$$\mathcal{B}(\boldsymbol{x}_c, r) = \left\{\boldsymbol{x} \,\middle|\, \|\boldsymbol{x} - \boldsymbol{x}_c\|_2^2 \leq r^2\right\}, \tag{2.6}$$

where $\boldsymbol{x}_c$ is the center of the ball and $r > 0$ is its radius. Examples of convex and nonconvex sets are depicted in Figure 2.1.

Next, we consider convex functions and convex optimization problems. A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is *convex* if its domain (denoted by $\operatorname{dom} f$) is a convex set and if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \operatorname{dom} f$ and any $\theta \in [0,1]$, we have

$$f\big(\theta \boldsymbol{x}_1 + (1-\theta)\boldsymbol{x}_2\big) \leq \theta f(\boldsymbol{x}_1) + (1-\theta) f(\boldsymbol{x}_2). \tag{2.7}$$

We call f *strictly convex* if strict inequality holds in (2.7) for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \operatorname{dom} f$ where

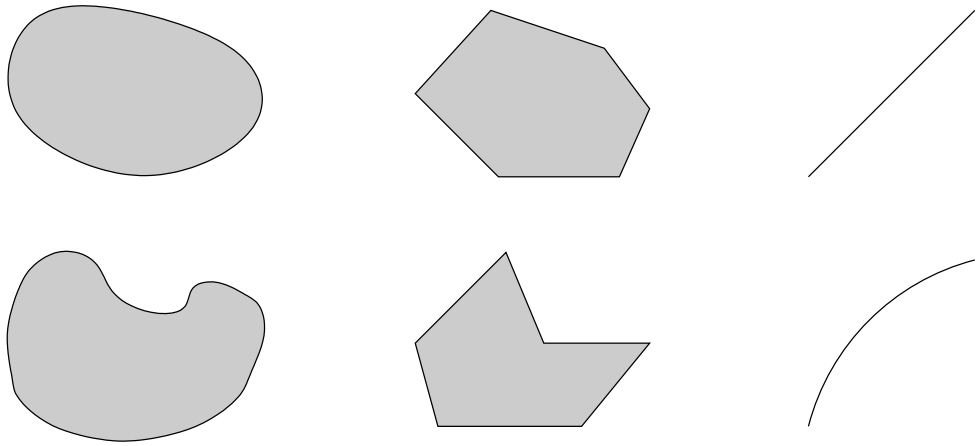


Figure 2.1: Some simple examples of convex sets (top row) and nonconvex sets (bottom row).

$\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ and $\theta \in (0,1)$. A function f is (strictly) *concave* if $-f$ is (strictly) convex. We say that f is *affine* if f is convex and concave. If f is differentiable, then the condition in (2.7) is equivalent to

$$f(\boldsymbol{x}_2) \geq f(\boldsymbol{x}_1) + \nabla f(\boldsymbol{x}_1)^{\mathrm{T}}(\boldsymbol{x}_2 - \boldsymbol{x}_1) \tag{2.8}$$

for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \operatorname{dom} f$. This is probably the most important property of convex functions. Suppose $\nabla f(\boldsymbol{x}_1) = \mathbf{0}$, then (2.8) tells us that $f(\boldsymbol{x}_2) \geq f(\boldsymbol{x}_1)$ for all $\boldsymbol{x}_2 \in \operatorname{dom} f$, i.e., f attains a global minimum at $\boldsymbol{x}_1$. The *α-sublevel set* of a function $f\colon \mathbb{R}^n \to \mathbb{R}$ is

$$\mathcal{S}_\alpha = \{\boldsymbol{x} \in \operatorname{dom} f | f(\boldsymbol{x}) \leq \alpha\}. \tag{2.9}$$

The sublevel sets of a convex function are convex for any $\alpha \in \mathbb{R}$. However, the converse is not true; a function with convex sublevel sets need not be convex. Similarly to (2.9), the *α-superlevel set* of f is

$$\mathcal{S}^\alpha = \{\boldsymbol{x} \in \operatorname{dom} f | f(\boldsymbol{x}) \geq \alpha\}. \tag{2.10}$$

If f is concave its superlevel sets are convex, but the converse is not true. A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is called *quasiconvex* if its domain and all its sublevel sets $\mathcal{S}_\alpha$ for $\alpha \in \mathbb{R}$ are convex. We call f *quasiconcave* if $-f$ is quasiconvex. A *quasilinear* function is both quasiconvex and quasiconcave. We note that every convex (concave) function is also quasiconvex (quasiconcave), but the converse is not true. Figure 2.2 depicts the graph of (a) a convex function, (b) a quasiconvex function, and (c) a nonconvex function.

An optimization problem of the form

$$\begin{aligned} &\min_{\boldsymbol{x}} \ f_0(\boldsymbol{x}) \\ &\text{subject to } \ \boldsymbol{x} \in \mathcal{X} \end{aligned} \tag{2.11}$$

is a *convex optimization problem* if the *objective function* f_0 and the *feasible set* $\mathcal{X}$ are

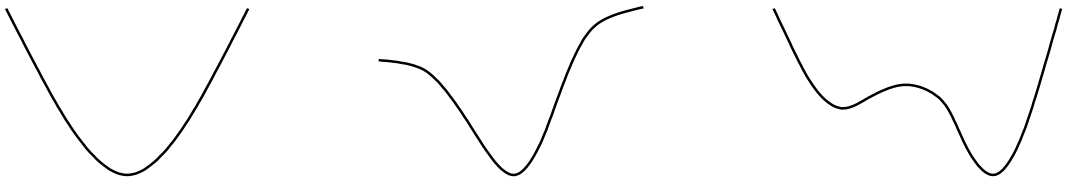


Figure 2.2: Graph of (a) a convex function, (b) a quasiconvex function, and (c) a nonconvex function.

convex. Writing the feasible set in (2.11) in terms of p equality constraints and m inequality constraints yields a convex optimization problem in standard form:

$$\begin{aligned} \min_{\boldsymbol{x}} \;& f_0(\boldsymbol{x}) \\ \text{subject to} \;& f_i(\boldsymbol{x}) \leq 0, \quad i = 1, \dots, m, \\ & h_j(\boldsymbol{x}) = 0, \quad j = 1, \dots, p. \end{aligned} \tag{2.12}$$

Here, the *inequality constraint functions* f_i, $i = 1, \dots, m$, are convex and the *equality constraint functions* h_j, $j = 1, \dots, p$, are affine. In this context we call $\boldsymbol{x} \in \mathbb{R}^n$ the *optimization variable*. We call $\boldsymbol{x}^\star$ a *(globally) optimal point* or simply *optimal* if $\boldsymbol{x}^\star \in \mathcal{X}$, i.e., $\boldsymbol{x}^\star$ is feasible, and $f_0(\boldsymbol{x}) \geq f_0(\boldsymbol{x}^\star)$ for all $\boldsymbol{x} \in \mathcal{X}$. A point $\boldsymbol{x}^*$ is *locally optimal* if there exists an $r > 0$ such that $f_0(\boldsymbol{x}) \geq f_0(\boldsymbol{x}^*)$ for all $\boldsymbol{x} \in \mathcal{B}(\boldsymbol{x}^*, r)$. A fundamental property of convex optimization problems is the fact that every local optimum is also a global optimum [12, Subsection 4.2.2].

We next state important inequalities related to convex functions. Let x be a random variable and let f be a convex function, then we have [12, Subsection 3.1.8]

$$f\big(\mathbb{E}\{\mathsf{x}\}\big) \leq \mathbb{E}\big\{f(\mathsf{x})\big\}, \tag{2.13}$$

provided that the expectations exist. The inequality in (2.13) is known as *Jensen's inequality*. An extension of Jensen's inequality holds for probability distributions which are compactly supported on an interval $[a, b] \subseteq \mathbb{R}$ and real-valued convex functions $f\colon [a, b] \to \mathbb{R}$. In this case we have

$$f\big(\mathbb{E}\{\mathsf{x}\}\big) \leq \mathbb{E}\big\{f(\mathsf{x})\big\} \leq f(a) + \frac{f(b) - f(a)}{b - a}\big(\mathbb{E}\{\mathsf{x}\} - a\big). \tag{2.14}$$

We note that the first inequality in (2.14) becomes an equality if x is constant, i.e., if $\mathsf{x} \equiv \mathbb{E}\{\mathsf{x}\}$, and we have equality in the second inequality if x takes the values a and b with probabilities $\varepsilon \in [0, 1]$ and $1 - \varepsilon$, respectively. A proof of (2.14) is given in [59, Chapter 3].

Finally, we introduce the concept of extreme points [10, Appendix B.4]. We call a point $\boldsymbol{x} \in \mathcal{S}$ an *extreme point* of a convex set $\mathcal{S}$ if $\boldsymbol{x}$ does not lie strictly within a line segment contained in $\mathcal{S}$. Equivalently, $\boldsymbol{x} \in \mathcal{S}$ is an extreme point if it cannot be expressed as a convex combination of vectors in $\mathcal{S}$ which are all different from $\boldsymbol{x}$. It turns out that any compact

convex set is equal to the convex hull of its extreme points. This result is known as the *Krein-Milman theorem* [53]. Furthermore, it can be shown that any nonempty, closed, and convex set has at least one extreme point if and only if it does not contain a line, i.e., a set of the form $\{\boldsymbol{x} + \alpha\boldsymbol{b} \,|\, \alpha \in \mathbb{R}\}$ with $\boldsymbol{b} \neq \boldsymbol{0}$.

2.2 Information Theory

Before we begin our discussion, we first introduce some notation and conventions. For our purposes it is sufficient to consider random variables which have a *probability density function* (pdf) or a *probability mass function* (pmf). We use $p_{\mathsf{x}}(x)$ to denote the pdf or pmf of a random variable x. It will be clear from the context whether $p_{\mathsf{x}}(x)$ is a pdf or a pmf. With some abuse of notation we write the pdf of a discrete random variable $\mathsf{x} \in \mathcal{X}$ as

$$p_{\mathsf{x}}(x) = \sum_{x' \in \mathcal{X}} \mathbb{P}\{\mathsf{x}{=}x'\}\delta(x - x'), \tag{2.15}$$

where $\delta(x)$ is the Dirac delta function. When there is no possibility for confusion, we simply write $p(x)$ instead of $p_{\mathsf{x}}(x)$. We use log to denote the *natural logarithm* (base e) and $\log_2$ denotes the *binary logarithm* to the base 2. Information-theoretic quantities are in *nats* if we use the natural logarithm and in *bits* if we use binary logarithm. We employ the usual conventions that $0 \log 0 = 0$, $0 \log \frac{0}{0} = 0$, $0 \log \frac{0}{q} = 0$, and $p \log \frac{p}{0} = \infty$ (for $p > 0$).

Next, we define basic information-theoretic quantities using the notation of [20]. The *entropy* of a discrete random variable $\mathsf{x} \in \mathcal{X}$ is

$$H(\mathsf{x}) \triangleq -\mathbb{E}\{\log p(\mathsf{x})\} = -\sum_{x \in \mathcal{X}} p(x) \log p(x). \tag{2.16}$$

We note that $H(\mathsf{x}) \geq 0$, where we have equality if x is constant, i.e., nonrandom. An upper bound on the entropy of x is $H(\mathsf{x}) \leq \log|\mathcal{X}|$, where we have equality if x is uniformly distributed over the set $\mathcal{X}$. We note that the entropy $H(\mathsf{x})$ is concave in $p(x)$. Let us consider the special case of a binary random variable $\mathsf{x} \in \{0, 1\}$ with $\mathbb{P}\{\mathsf{x}{=}1\} = p$. In this case we have $H(\mathsf{x}) = -p \log p - (1-p)\log(1-p)$. It is convenient to write this entropy as a function of p, yielding the *binary entropy function* (in bits)

$$h_2(p) \triangleq -p \log_2 p - (1-p)\log_2(1-p). \tag{2.17}$$

Entropy naturally extends to two random variables $\mathsf{x} \in \mathcal{X}$ and $\mathsf{y} \in \mathcal{Y}$ which can be considered as one vector-valued random variable. The *joint entropy* $H(\mathsf{x}, \mathsf{y})$ is defined as

$$H(\mathsf{x}, \mathsf{y}) \triangleq -\mathbb{E}\{\log p(\mathsf{x}, \mathsf{y})\} = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y). \tag{2.18}$$

We note that the joint entropy in (2.18) can be rewritten as follows:

$$H(\mathsf{x},\mathsf{y}) = -\mathbb{E}\{\log p(\mathsf{x})\} - \mathbb{E}\{\log p(\mathsf{y}|\mathsf{x})\} = H(\mathsf{x}) + H(\mathsf{y}|\mathsf{x}). \tag{2.19}$$

This relation is known as the *chain rule for entropy*. The last term in (2.19) is the *conditional entropy* of y given x:

$$H(\mathsf{y}|\mathsf{x}) \triangleq -\mathbb{E}\{\log p(\mathsf{y}|\mathsf{x})\} = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y|x). \tag{2.20}$$

We note that $H(\mathsf{x},\mathsf{y}) \leq H(\mathsf{x}) + H(\mathsf{y})$ or, equivalently, $H(\mathsf{y}|\mathsf{x}) \leq H(\mathsf{y})$, where we have equality if and only if x and y are statistically independent.

The *relative entropy* (or Kullback-Leibler divergence) between two probability distributions $p(x)$ and $q(x)$ which are defined over the same set $\mathcal{X}$ is

$$D\big(p(x)\|q(x)\big) \triangleq \mathbb{E}_p\left\{\log\frac{p(\mathsf{x})}{q(\mathsf{x})}\right\} = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}, \tag{2.21}$$

where $\mathbb{E}_p$ denotes expectation with respect to $p(x)$. Relative entropy is nonnegative, i.e., we have

$$D\big(p(x)\|q(x)\big) \geq 0, \tag{2.22}$$

where we have equality in (2.22) if $p(x) = q(x)$. This inequality is known as the *information inequality* and its proof is based on Jensen's inequality (cf. (2.13)). The relative entropy $D\big(p(x)\|q(x)\big)$ is convex in $p(x)$ and $q(x)$.

The *mutual information* $I(\mathsf{x};\mathsf{y})$ between the random variables $\mathsf{x}\in\mathcal{X}$ and $\mathsf{y}\in\mathcal{Y}$ is defined as

$$I(\mathsf{x};\mathsf{y}) \triangleq \mathbb{E}\left\{\log\frac{p(\mathsf{x},\mathsf{y})}{p(\mathsf{x})p(\mathsf{y})}\right\} = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}. \tag{2.23}$$

It is not hard to see that $I(\mathsf{x};\mathsf{y}) = D\big(p(x,y)\|p(x)p(y)\big)$. Due to (2.22), we can therefore conclude that $I(\mathsf{x};\mathsf{y}) \geq 0$ with equality if and only if x and y are statistically independent. The mutual information $I(\mathsf{x};\mathsf{y})$ is concave in $p(x)$ for fixed $p(y|x)$. For fixed $p(x)$, $I(\mathsf{x};\mathsf{y})$ is convex in $p(y|x)$. Rewriting $I(\mathsf{x};\mathsf{y})$ in terms of entropies yields

$$I(\mathsf{x};\mathsf{y}) = H(\mathsf{x}) - H(\mathsf{x}|\mathsf{y}) \tag{2.24a}$$
$$= H(\mathsf{y}) - H(\mathsf{y}|\mathsf{x}) \tag{2.24b}$$
$$= H(\mathsf{x}) + H(\mathsf{y}) - H(\mathsf{x},\mathsf{y}). \tag{2.24c}$$

The relations in (2.24) are depicted in Figure 2.3 using a Venn diagram. Furthermore, we note that $I(\mathsf{x};\mathsf{x}) = H(\mathsf{x})$, i.e., entropy is "self-information". The *chain rule of mutual information* reads

$$I(\mathsf{x}_1,\mathsf{x}_2;\mathsf{y}) = I(\mathsf{x}_1;\mathsf{y}) + I(\mathsf{x}_2;\mathsf{y}|\mathsf{x}_1), \tag{2.25}$$

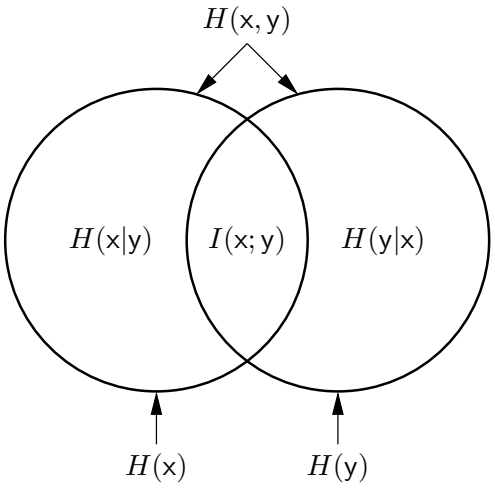


Figure 2.3: Relationship between entropy and mutual information.

where the last term in (2.25) is the *conditional mutual information* of x_2 and y given x_1:

$$I(\mathsf{x}_2;\mathsf{y}|\mathsf{x}_1) \triangleq \mathbb{E}\left\{\log \frac{p(\mathsf{x}_2,\mathsf{y}|\mathsf{x}_1)}{p(\mathsf{x}_2|\mathsf{x}_1)p(\mathsf{y}|\mathsf{x}_1)}\right\} = H(\mathsf{x}_2|\mathsf{x}_1) - H(\mathsf{x}_2|\mathsf{x}_1,\mathsf{y}). \tag{2.26}$$

So far we have only considered discrete random variables. When considering continuous random variables, the definitions and properties of relative entropy and mutual information remain unchanged. However, we have to pay particular attention when studying the entropy of continuous random variables which is called *differential entropy*. The differential entropy $h(\mathsf{x})$ of a continuous random variable with pdf $p(x)$ supported on the set $\mathcal{X}$ is defined as

$$h(\mathsf{x}) \triangleq -\int_{\mathcal{X}} p(x)\log p(x)\mathrm{d}x. \tag{2.27}$$

It is important to note that the integral in (2.27) may not exist and, if it exists, it need not be nonnegative. Joint differential entropy and conditional differential entropy are defined analogously to the discrete case. As an example let us consider an n-dimensional Gaussian random vector with mean vector $\boldsymbol{\mu}_{\mathbf{x}}$ and covariance matrix $\boldsymbol{C}_{\mathbf{x}}$, i.e., $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{C}_{\mathbf{x}})$. In this case we have [20, Theorem 8.4.1]

$$h(\mathbf{x}) = \frac{1}{2}\log\big((2\pi e)^n \det \boldsymbol{C}_{\mathbf{x}}\big). \tag{2.28}$$

The fact that conditioning reduces entropy holds also for differential entropy, i.e., we have $h(\mathsf{y}|\mathsf{x}) \leq h(\mathsf{y})$. Mutual information can be expressed in terms of differential entropies as in (2.24) (of course, with $H(\cdot)$ replaced by $h(\cdot)$) as long as the respective differential entropies are finite.

Finally, we introduce the notion of Markov chains and discuss the data processing inequality. The random variables $\mathsf{x},\mathsf{y},\mathsf{z}$ are said to form a *Markov chain*, denoted by $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z}$, if

x and z are statistically independent given y, i.e., if

$$p(x, z|y) = p(x|y)p(z|y). \tag{2.29}$$

We note that (2.29) is equivalent to

$$p(x, y, z) = p(z|y)p(y|x)p(x) = p(x|y)p(y|z)p(z), \tag{2.30}$$

and therefore $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z}$ implies $\mathsf{z} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{x}$. Given a Markov chain $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z}$, the *data processing inequality* states that $I(\mathsf{x};\mathsf{y}) \geq I(\mathsf{x};\mathsf{z})$. This can be seen by writing

$$I(\mathsf{x};\mathsf{y},\mathsf{z}) = I(\mathsf{x};\mathsf{z}) + I(\mathsf{x};\mathsf{y}|\mathsf{z}) = I(\mathsf{x};\mathsf{y}) + I(\mathsf{x};\mathsf{z}|\mathsf{y}) \tag{2.31}$$

and by noting that $I(\mathsf{x};\mathsf{z}|\mathsf{y}) = 0$ due to (2.29). Similarly, (2.31) implies that $I(\mathsf{x};\mathsf{y}) \geq I(\mathsf{x};\mathsf{y}|\mathsf{z})$. For $\mathsf{z} = g(\mathsf{y})$ the data processing inequality implies $H(\mathsf{x}|\mathsf{y}) \leq H(\mathsf{x}|g(\mathsf{y}))$ with equality if the function $g(\cdot)$ is invertible. Therefore, processing y can only increase the uncertainty about x.

2.3 Parameter Estimation

We consider "classical" estimation (cf., e.g., [50,82,99]) of a scalar *parameter* $\theta \in \Theta \subseteq \mathbb{R}$ from observed *data* $\boldsymbol{x} = (x_1 \cdots x_n)^{\mathrm{T}} \in \mathcal{X}$. In classical estimation, the parameter θ is deterministic and the data is distributed according to the pdf[1] $p(\boldsymbol{x};\theta)$. The *estimator* $\hat{\theta}(\boldsymbol{x})$ is a function of the data which is designed to minimize the *estimation error* $\hat{\theta} - \theta$ in some suitable sense. Throughout, we use the mean-square error (MSE)

$$\mathrm{MSE}_{\hat{\theta}}(\theta) \triangleq \mathbb{E}\{(\hat{\theta}(\mathsf{x}) - \theta)^2\} = \int_{\mathcal{X}} (\hat{\theta}(\boldsymbol{x}) - \theta)^2 p(\boldsymbol{x};\theta)\mathrm{d}\boldsymbol{x} \tag{2.32}$$

as performance metric. It is important to note that the MSE depends on θ (although we usually do not make this dependence explicit for the sake of notational simplicity). Since the MSE in (2.32) is the mean power of the estimation error, we have the following decomposition:

$$\mathrm{MSE}_{\hat{\theta}} = \mathrm{var}\{\hat{\theta} - \theta\} + \big(\mathbb{E}\{\hat{\theta}(\mathsf{x}) - \theta\}\big)^2 = \mathrm{var}\{\hat{\theta}\} + \mathrm{bias}^2\{\hat{\theta}\}, \tag{2.33}$$

where

$$\mathrm{var}\{\hat{\theta}\} = \mathrm{var}\{\hat{\theta} - \theta\} = \mathbb{E}\big\{(\hat{\theta}(\mathsf{x}) - \mathbb{E}\{\hat{\theta}(\mathsf{x})\})^2\big\}, \tag{2.34}$$

and

$$\mathrm{bias}\{\hat{\theta}\} = \mathbb{E}\{\hat{\theta}(\mathsf{x}) - \theta\} = \mathbb{E}\{\hat{\theta}(\mathsf{x})\} - \theta \tag{2.35}$$

[1] The notation $p(\boldsymbol{x};\theta)$ indicates that the distribution of the random vector x is parametrized by θ, i.e., each value of θ corresponds to one distribution of the data. We note that $p(\boldsymbol{x};\theta)$ should not be confused with a joint pdf or a conditional pdf (which would not make sense since θ is deterministic).

denote the variance and the bias of the estimator $\hat{\theta}$, respectively. We say that the estimator $\hat{\theta}_1$ *dominates* the estimator $\hat{\theta}_2$ if and only if $\mathrm{MSE}_{\hat{\theta}_1}(\theta) \leq \mathrm{MSE}_{\hat{\theta}_2}(\theta)$ for all $\theta \in \Theta$. An estimator is said to be *unbiased* if $\mathbb{E}\{\hat{\theta}(\mathbf{x})\} = \theta$ for all $\theta \in \Theta$. We note that for a given estimation problem the existence of an unbiased estimator is not guaranteed. An estimator (more precisely, a sequence of estimators) is called *consistent* if it converges (in probability) to the true parameter as the number of data points tends to infinity, i.e., if

$$\lim_{n\to\infty} \hat{\theta}_n(\boldsymbol{x}_n) = \theta, \quad \forall \theta \in \Theta. \tag{2.36}$$

Here, $\hat{\theta}_n$ denotes the estimator which operates on the length-n data set $\boldsymbol{x}_n = (x_1 \cdots x_n)^{\mathrm{T}}$. In contrast to the unbiasedness of an estimator, consistency is an asymptotic property. Consistency does not imply unbiasedness and vice versa. However, an estimator is consistent if and only if it converges and it is asymptotically unbiased (i.e., $\lim_{n\to\infty} \hat{\theta}_n$ is unbiased).

A lower bound on the variance of an unbiased estimator of a parameter θ is given by the Cramér-Rao lower bound (CRLB) [21, 84]. Under mild regularity conditions on $p(\boldsymbol{x};\theta)$ and $\hat{\theta}(\boldsymbol{x})$, the CRLB states that

$$\mathrm{MSE}_{\hat{\theta}}(\theta) = \mathrm{var}\{\hat{\theta}\} \geq J^{-1}(\theta), \tag{2.37}$$

where $J(\theta)$ denotes the *Fisher information* which is defined as

$$J(\theta) \triangleq \mathbb{E}\left\{ \left(\frac{\partial}{\partial\theta} \log p(\mathbf{x};\theta) \right)^2 \right\}. \tag{2.38}$$

An important property of the Fisher information is that it is *additive* for independent data samples. Therefore, if we have n independent and identically distributed observations, then the CRLB is $1/n$ times the CRLB for a single observation. For a parameter ψ which is related to θ by a continuously differentiable function $g(\cdot)$ such that $\theta = g(\psi)$, the Fisher information can be written as follows:

$$J_\psi(\psi) = J_\theta\big(g(\psi)\big) \left(\frac{\mathrm{d}}{\mathrm{d}\psi} g(\psi) \right)^2. \tag{2.39}$$

Here, J_θ and J_ψ denote the Fisher information of θ and ψ, respectively. In terms of ψ, the CRLB (2.37) for $\theta(\psi)$ can thus be written as

$$\mathrm{MSE}_{\hat{\theta}}\big(\theta(\psi)\big) = \mathrm{var}\{\hat{\theta}\} \geq J_\psi^{-1}(\psi) \left(\frac{\mathrm{d}}{\mathrm{d}\psi} g(\psi) \right)^2. \tag{2.40}$$

An unbiased estimator is called *efficient* if its MSE attains the CRLB, i.e., if $\mathrm{MSE}_{\hat{\theta}}(\theta) = 1/J(\theta)$ for all $\theta \in \Theta$. We note the CRLB need not be tight, that is, an efficient estimator may not exist. In fact, an efficient estimator $\hat{\theta}_{\mathrm{eff}}(\boldsymbol{x})$ exists if and only if $\frac{\partial}{\partial\theta} \log f(\boldsymbol{x};\theta)$ can be written as [50, Section 3.4]

$$\frac{\partial}{\partial\theta} \log f(\boldsymbol{x};\theta) = J(\theta)\big(\hat{\theta}_{\mathrm{eff}}(\boldsymbol{x}) - \theta\big). \tag{2.41}$$

If an efficient estimator exists, it is the *minimum-variance unbiased* (MVU) estimator. The MVU estimator is the estimator which uniformly minimizes the variance among all unbiased estimators for all values of θ. However, an MVU estimator need not exist and, if it exists, it need not be an efficient estimator. In general it is hard to find an MVU estimator and the approach using (2.41) can only be used if an efficient estimator exists. In case an efficient estimator does not exist, we can use the *Rao-Blackwell-Lehmann-Scheffé* theorem (cf. [50, Theorem 5.2]) to find an MVU estimator (assuming it exists).

We next introduce the concept of (complete) sufficient statistics. Loosely speaking, a function $T(\boldsymbol{x})$ of the data $\boldsymbol{x}$ is called a *sufficient statistic* if $T(\boldsymbol{x})$ contains all information about θ that is contained in $\boldsymbol{x}$. More precisely, a statistic $T(\boldsymbol{x})$ is sufficient if and only if the conditional distribution $p(\boldsymbol{x}|T(\boldsymbol{x});\theta)$ does not depend on θ. While this condition may be difficult to check in practice, we can alternatively use the *Neyman-Fisher factorization* theorem which is usually easier to apply. The Neyman-Fisher factorization theorem provides the following sufficient and necessary condition for the sufficiency of a statistic $T(\boldsymbol{x})$.

Theorem 2.1 (cf. [50, Theorem 5.1])**.** *If we can factor $p(\boldsymbol{x};\theta)$ as*

$$p(\boldsymbol{x};\theta) = g(\boldsymbol{x})h(T(\boldsymbol{x}),\theta), \tag{2.42}$$

where g depends only on $\boldsymbol{x}$ and h depends on $\boldsymbol{x}$ only through $T(\boldsymbol{x})$, then $T(\boldsymbol{x})$ is a sufficient statistic for θ. Conversely, if $T(\boldsymbol{x})$ is a sufficient statistic for θ, then $p(\boldsymbol{x};\theta)$ can be factored as in (2.42).

A sufficient statistic is said to be *complete* if there exists a *unique* function g such that $g\big(T(\boldsymbol{x})\big)$ is unbiased, i.e., $\mathbb{E}\big\{g\big(T(\boldsymbol{x})\big)\big\} = \theta$ for all $\theta \in \Theta$. Assuming that there exist two functions g_1 and g_2 such that $g_1\big(T(\boldsymbol{x})\big)$ and $g_2\big(T(\boldsymbol{x})\big)$ are unbiased and letting $h\big(T(\boldsymbol{x})\big) = g_2\big(T(\boldsymbol{x})\big) - g_1\big(T(\boldsymbol{x})\big)$ yields

$$\mathbb{E}\{h\big(T(\mathbf{x})\big)\} = \int_{\mathcal{X}} h\big(T(\boldsymbol{x})\big)p(\boldsymbol{x};\theta)\mathrm{d}\boldsymbol{x} = 0, \quad \forall\theta \in \Theta. \tag{2.43}$$

If $h \equiv 0$ is the only function that fulfills (2.43), then $T(\boldsymbol{x})$ is a complete sufficient statistic. Conversely, if $T(\boldsymbol{x})$ is a complete sufficient statistic, then $h \equiv 0$ (due to uniqueness) and (2.43) is satisfied.

2.4 Hypothesis Testing

In this section, we discuss simple binary and m-ary Bayesian hypothesis tests (cf., e.g., [51, 82, 99]). Consider a source which produces an output $\mathcal{H}$ which is either $\mathcal{H}_0$ or $\mathcal{H}_1$. The source output is mapped probabilistically to an observation $\boldsymbol{x} \in \mathcal{X}$ via the conditional pdfs $p(\boldsymbol{x}|\mathcal{H}_0)$ and $p(\boldsymbol{x}|\mathcal{H}_1)$. In the Bayesian setting, the source output is random and the prior probabilities $\mathbb{P}\{\mathcal{H} = \mathcal{H}_0\}$ and $\mathbb{P}\{\mathcal{H} = \mathcal{H}_1\}$ are known. We refer to $\mathcal{H}_0$ and $\mathcal{H}_1$ as *hypotheses*

and the task of a *hypothesis test* is to infer from the observation $\boldsymbol{x}$ which hypothesis is in force. Depending on the domain, hypothesis testing is also known as signal detection (e.g., in radar applications) and, consequently, a hypothesis test is also called a detector. Roughly speaking, a hypothesis testing problem can be viewed as an estimation problem with a finite parameter set.

A hypothesis test is called *simple* if the probability distribution of the observation under each hypothesis is fully known, i.e., if $p(\boldsymbol{x}|\mathcal{H}_0)$ and $p(\boldsymbol{x}|\mathcal{H}_1)$ do not depend on any unknown parameters. A binary test can be represented by a *test function* (or *decision rule*, or *detector*) of the form

$$\phi(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in \mathcal{X}_0 \\ 1, & \boldsymbol{x} \in \mathcal{X}_1 \end{cases}, \tag{2.44}$$

where the output of the test function corresponds to the index of the accepted hypothesis. The set $\mathcal{X}_0$ is called *acceptance region* and $\mathcal{X}_1$ is the *critical region* (or *rejection region*), where $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$ and $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$. The test in (2.44) accepts $\mathcal{H}_0$ and rejects $\mathcal{H}_1$ if $\boldsymbol{x} \in \mathcal{X}_0$; it rejects $\mathcal{H}_0$ and therefore accepts $\mathcal{H}_1$ if $\boldsymbol{x} \in \mathcal{X}_1$.

The probabilities $\mathbb{P}\{\phi(\mathsf{x}) = i|\mathcal{H} = \mathcal{H}_j\}$, $i, j \in \{0, 1\}$ are important performance measures for a binary test. In particular, we have the following four probabilities:

1. The acceptance probability, i.e., the probability of correctly accepting $\mathcal{H}_0$, is

$$P_A \triangleq \mathbb{P}\{\phi(\mathsf{x}) = 0|\mathcal{H} = \mathcal{H}_0\} = \int_{\mathcal{X}} \big(1 - \phi(\boldsymbol{x})\big) p(\boldsymbol{x}|\mathcal{H}_0) \mathrm{d}\boldsymbol{x} = \mathbb{E}\{1 - \phi(\mathsf{x})|\mathcal{H} = \mathcal{H}_0\}. \tag{2.45}$$

2. The *false alarm probability*, i.e., the probability of incorrectly accepting $\mathcal{H}_1$, is

$$P_F \triangleq \mathbb{P}\{\phi(\mathsf{x}) = 1|\mathcal{H} = \mathcal{H}_0\} = \int_{\mathcal{X}} \phi(\boldsymbol{x}) p(\boldsymbol{x}|\mathcal{H}_0) \mathrm{d}\boldsymbol{x} = \mathbb{E}\{\phi(\mathsf{x})|\mathcal{H} = \mathcal{H}_0\}. \tag{2.46}$$

   The probability P_F is sometimes called the *size* of a test and the corresponding error event is referred to as *type I error* or simply *false alarm*.

3. The *detection probability* (or *power*), i.e., the probability of correctly accepting $\mathcal{H}_1$, is

$$P_D \triangleq \mathbb{P}\{\phi(\mathsf{x}) = 1|\mathcal{H} = \mathcal{H}_1\} = \int_{\mathcal{X}} \phi(\boldsymbol{x}) p(\boldsymbol{x}|\mathcal{H}_1) \mathrm{d}\boldsymbol{x} = \mathbb{E}\{\phi(\mathsf{x})|\mathcal{H} = \mathcal{H}_1\}. \tag{2.47}$$

4. The miss probability, i.e., the probability of incorrectly accepting $\mathcal{H}_0$, is

$$P_M \triangleq \mathbb{P}\{\phi(\mathsf{x}) = 0|\mathcal{H} = \mathcal{H}_1\} = \int_{\mathcal{X}} \big(1 - \phi(\boldsymbol{x})\big) p(\boldsymbol{x}|\mathcal{H}_1) \mathrm{d}\boldsymbol{x} = \mathbb{E}\{1 - \phi(\mathsf{x})|\mathcal{H} = \mathcal{H}_1\}. \tag{2.48}$$

   The error event corresponding to P_M is called *type II error* or simply *miss*.

The probabilities in (2.45)-(2.48) are not independent of each other since we have $P_A + P_F = 1$ and $P_D + P_M = 1$. Therefore, we can restrict our attention to, say, P_F and P_D. Obviously, we would like to find tests with small P_F and large P_D. However, changing P_F by modifying

$\phi(\boldsymbol{x})$ will simultaneously change P_D. This raises two questions: when is a test optimal and how can we implement an optimal test?

To this end, we first consider the Bayesian risk. Let $C_{ij} \geq 0$ denote the cost of deciding in favor of $\mathcal{H}_i$ when $\mathcal{H}_j$ is in force. The cost of a Bayesian test ϕ can then be written as

$$\mathsf{C}(\phi) = \begin{cases} C_{00}, & \phi(\mathbf{x}) = 0,\ \mathcal{H} = \mathcal{H}_0 \\ C_{01}, & \phi(\mathbf{x}) = 0,\ \mathcal{H} = \mathcal{H}_1 \\ C_{10}, & \phi(\mathbf{x}) = 1,\ \mathcal{H} = \mathcal{H}_0 \\ C_{11}, & \phi(\mathbf{x}) = 1,\ \mathcal{H} = \mathcal{H}_1 \end{cases}. \tag{2.49}$$

The cost $\mathsf{C}(\phi)$ in (2.49) is a random variable because $\mathbf{x}$ and $\mathcal{H}$ are random. The *Bayesian risk* $R(\phi)$ associated to a test function ϕ is the expected value of $\mathsf{C}(\phi)$, i.e., we have

$$R(\phi) \triangleq \mathbb{E}\{\mathsf{C}(\phi)\} = \sum_{j=0}^{1}\sum_{i=0}^{1} C_{ij}\mathbb{P}\{\phi(\mathbf{x}) = i, \mathcal{H} = \mathcal{H}_j\} = \sum_{j=0}^{1} R_j(\phi)\mathbb{P}\{\mathcal{H} = \mathcal{H}_j\}, \tag{2.50}$$

where $R_j(\phi)$ denotes the conditional risk

$$R_j(\phi) \triangleq \mathbb{E}\{\mathsf{C}(\phi)|\mathcal{H} = \mathcal{H}_j\} = \sum_{i=0}^{1} C_{ij}\mathbb{P}\{\phi(\mathbf{x}) = i|\mathcal{H} = \mathcal{H}_j\}. \tag{2.51}$$

In the special case $C_{ij} = 1 - \delta_{i,j}$ we have

$$R(\phi) = P_F\mathbb{P}\{\mathcal{H} = \mathcal{H}_0\} + P_M\mathbb{P}\{\mathcal{H} = \mathcal{H}_1\}, \tag{2.52}$$

i.e., in this case the Bayesian risk equals the probability of making a wrong decision. For a given cost assignment, a test function ϕ is optimal in the Bayesian sense if it minimizes the Bayesian risk $R(\phi)$ among all test functions. The optimal test ϕ_B is thus given by [99, Section 2.2]

$$\phi_B = \arg\min_{\phi} R(\phi). \tag{2.53}$$

To find ϕ_B, we rewrite the Bayesian risk in (2.53) as follows:

$$\phi_B = \arg\min_{\phi} \sum_{j=0}^{1}\sum_{i=0}^{1} C_{ij}\mathbb{P}\{\phi(\mathbf{x}) = i|\mathcal{H} = \mathcal{H}_j\}\mathbb{P}\{\mathcal{H} = \mathcal{H}_j\} \tag{2.54}$$

$$= \arg\min_{\phi} \sum_{j=0}^{1} \big[C_{0j}\mathbb{P}\{\phi(\mathbf{x}) = 0|\mathcal{H} = \mathcal{H}_j\} + C_{1j}\mathbb{P}\{\phi(\mathbf{x}) = 1|\mathcal{H} = \mathcal{H}_j\}\big]\mathbb{P}\{\mathcal{H} = \mathcal{H}_j\} \tag{2.55}$$

$$= \arg\min_{\phi} \sum_{j=0}^{1} (C_{1j} - C_{0j})\mathbb{P}\{\phi(\mathbf{x}) = 1|\mathcal{H} = \mathcal{H}_j\}\mathbb{P}\{\mathcal{H} = \mathcal{H}_j\} \tag{2.56}$$

$$= \arg\min_{\phi} \int_{\mathcal{X}} \phi(\boldsymbol{x}) \sum_{j=0}^{1} (C_{1j} - C_{0j}) p(\boldsymbol{x}|\mathcal{H} = \mathcal{H}_j)\mathbb{P}\{\mathcal{H} = \mathcal{H}_j\}\mathrm{d}\boldsymbol{x}. \tag{2.57}$$

The integrand in (2.57) can be minimized separately for each $\boldsymbol{x} \in \mathcal{X}$ since all quantities except $C_{1j} - C_{0j}$ are nonnegative. To minimize the Bayesian risk we have to set $\phi(\boldsymbol{x}) = 0$ if $\sum_{j=0}^{1}(C_{1j} - C_{0j})p(\boldsymbol{x}|\mathcal{H}=\mathcal{H}_j)\mathbb{P}\{\mathcal{H}=\mathcal{H}_j\}$ is positive and $\phi(\boldsymbol{x}) = 1$ otherwise. Thus, under the assumption that $C_{10} > C_{00}$ and $C_{01} > C_{11}$, the optimal test function is given by[2]

$$\phi_B(\boldsymbol{x}) = \begin{cases} 0, & L(\boldsymbol{x}) > \gamma \\ 1, & L(\boldsymbol{x}) \leq \gamma \end{cases}, \tag{2.58}$$

where $L(\boldsymbol{x})$ is the likelihood ratio[3], which is defined as

$$L(\boldsymbol{x}) \triangleq \frac{p(\boldsymbol{x}|\mathcal{H}=\mathcal{H}_0)}{p(\boldsymbol{x}|\mathcal{H}=\mathcal{H}_1)}, \tag{2.59}$$

and the threshold γ equals

$$\gamma = \frac{C_{01} - C_{11}}{C_{10} - C_{00}} \frac{\mathbb{P}\{\mathcal{H}=\mathcal{H}_1\}}{\mathbb{P}\{\mathcal{H}=\mathcal{H}_0\}}. \tag{2.60}$$

This shows that the optimal test is a *likelihood ratio test* and ϕ_B can be implemented by computing $L(\boldsymbol{x})$ and comparing it to a threshold. It is important to note that independent of the number of data samples, the decision is always based on the scalar *test statistic* $L(\boldsymbol{x})$. Since $L(\boldsymbol{x})$ comprises all information that $\boldsymbol{x}$ carries about $\mathcal{H}$, the likelihood ratio is a *sufficient statistic* (cf. Theorem 2.1) for $\mathcal{H}$. For the special case $C_{ij} = 1 - \delta_{i,j}$ we have

$$\phi_B(\boldsymbol{x}) = \arg\max_{i \in \{0,1\}} \mathbb{P}\{\mathcal{H}=\mathcal{H}_i|\mathbf{x}=\boldsymbol{x}\}. \tag{2.61}$$

Hence, in this case the optimal detector is a maximum *a posteriori* (MAP) detector which minimizes the error probability (2.52). We note that it suffices to compute one posterior probability in (2.61), since $\mathbb{P}\{\mathcal{H}=\mathcal{H}_1|\mathbf{x}=\boldsymbol{x}\} = 1 - \mathbb{P}\{\mathcal{H}=\mathcal{H}_0|\mathbf{x}=\boldsymbol{x}\}$.

Next, we discuss the extension to simple m-ary Bayesian hypothesis tests. In this case we have m hypotheses $\mathcal{H}_0, \ldots, \mathcal{H}_{m-1}$ with the associated probabilistic mappings $p(\boldsymbol{x}|\mathcal{H}_i)$ and the prior probabilities $\mathbb{P}\{\mathcal{H} = \mathcal{H}_i\}$, $i = 0, \ldots, m-1$. There are m events corresponding to correct decisions and $m^2 - m$ error events and therefore the cost assignment consists of m^2 values $C_{ij} \geq 0$, $i, j \in \{0, \ldots, m-1\}$. The test function $\phi(\boldsymbol{x})$ partitions the observation space $\mathcal{X}$ into m disjoint decision regions $\mathcal{X}_i$ and we have $\phi(\boldsymbol{x}) = i$ if $\boldsymbol{x} \in \mathcal{X}_i$, $i = 0, \ldots, m-1$. As before, the value of the test function corresponds to the index of the accepted hypothesis. The Bayesian risk is defined analogously to (2.50) and the optimal Bayesian test is given by (2.53). In this case, rewriting the Bayesian risk yields

$$\phi_B = \arg\min_{\phi} \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} C_{ij} \mathbb{P}\{\phi(\mathbf{x})=i|\mathcal{H}=\mathcal{H}_j\}\mathbb{P}\{\mathcal{H}=\mathcal{H}_j\} \tag{2.62}$$

[2]We note that if $L(\boldsymbol{x}) = \gamma$, the value of $\phi_B(\boldsymbol{x}) \in \{0, 1\}$ is immaterial for the Bayesian risk. Therefore, we do not need to consider randomized test functions.

[3]We note that some authors define the likelihood ratio as $L(\boldsymbol{x}) = p(\boldsymbol{x}|\mathcal{H}=\mathcal{H}_1)/p(\boldsymbol{x}|\mathcal{H}=\mathcal{H}_0)$. This is an arbitrary choice without any fundamental consequences. The only difference is that the values of the optimal test function (2.58) are swapped and γ is replaced by γ^{-1}.

$$= \arg\min_{\phi} \sum_{i=0}^{m-1} \int_{\mathcal{X}} \mathbb{1}\{\phi(\boldsymbol{x}) = i\} \sum_{j=0}^{m-1} C_{ij} p(\boldsymbol{x}|\mathcal{H}=\mathcal{H}_j)\mathbb{P}\{\mathcal{H}=\mathcal{H}_j\}\mathrm{d}\boldsymbol{x} \tag{2.63}$$

$$= \arg\min_{\phi} \sum_{i=0}^{m-1} \int_{\mathcal{X}} \mathbb{1}\{\phi(\boldsymbol{x}) = i\} \sum_{j=0}^{m-1} C_{ij} \mathbb{P}\{\mathcal{H}=\mathcal{H}_j|\mathbf{x}=\boldsymbol{x}\}p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{2.64}$$

$$= \arg\min_{\phi} \sum_{i=0}^{m-1} \int_{\mathcal{X}} \mathbb{1}\{\phi(\boldsymbol{x}) = i\} \tilde{R}_i(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{2.65}$$

Here, $\mathbb{1}\{\cdot\}$ denotes the indicator function and $\tilde{R}_i(\boldsymbol{x}) = \sum_{j=0}^{m-1} C_{ij}\mathbb{P}\{\mathcal{H} = \mathcal{H}_j|\mathbf{x} = \boldsymbol{x}\}$ is the posterior risk of deciding for $\mathcal{H}_i$ given that $\boldsymbol{x}$ was observed. From (2.65) it can be seen that the Bayesian risk is minimized by setting $\phi(\boldsymbol{x}) = i$ if i is the index of the smallest posterior risk $\tilde{R}_i(\boldsymbol{x})$. We thus have

$$\phi_B(\boldsymbol{x}) = \arg\min_{i\in\{0,\dots,m-1\}} \tilde{R}_i(\boldsymbol{x}) \tag{2.66}$$

for the optimal Bayesian test function. The special case $C_{ij} = 1 - \delta_{i,j}$ minimizes the error probability and again yields a MAP detector. In this case we have

$$\phi_B(\boldsymbol{x}) = \arg\max_{i\in\{0,\dots,m-1\}} \mathbb{P}\{\mathcal{H}=\mathcal{H}_j|\mathbf{x}=\boldsymbol{x}\}. \tag{2.67}$$

For the special case $m = 2$, (2.66) and (2.67) simplify to (2.58) and (2.61), respectively.

2.5 Factor Graphs and the Sum-Product Algorithm

Let us consider functions $f(\boldsymbol{x})$ which (nontrivially) factor as

$$f(\boldsymbol{x}) = \prod_{k=1}^{K} f_k(\boldsymbol{x}_k), \tag{2.68}$$

where $\boldsymbol{x} = (x_1 \cdots x_n)^{\mathrm{T}}$. Each individual factor f_k depends on a subset of all variables denoted by $\boldsymbol{x}_k$. Factor graphs [54] are undirected bipartite graphs which allow us to graphically represent factorizations of the form (2.68). The vertices in a factor graph are called *variable nodes* (usually depicted by circles) and *factor nodes* (or function nodes; usually depicted by squares). A factor graph contains one vertex for each variable and one for each factor. Therefore, the factor graph corresponding to the factorization in (2.68) consists of n variable nodes and K factor nodes. A factor node is connected to a variable node by an undirected edge if the corresponding factor depends on that variable. As an example, Figure 2.4 depicts the factor graph corresponding to the factorization

$$f(x_1, x_2, x_3) = f_1(x_1)f_2(x_2)f_3(x_1, x_2, x_3). \tag{2.69}$$

Each assignment of values to the variables is called a *configuration* and the set of all possible configurations is the *configuration space*.

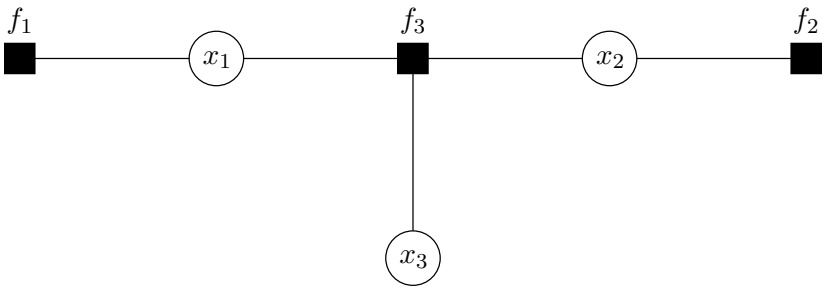


Figure 2.4: Factor graph corresponding to the factorization in (2.69).

In what follows, we focus on the case where the left-hand side of (2.68) is a probability distribution. In this case, factor graphs allow us to efficiently compute for example marginal distributions and maximum probability configurations. A key ingredient in the efficient computation of marginal distributions is the distributive law $a(b+c) = ab + ac$. Let us consider the computation of

$$f(x_1) = \sum_{\sim x_1} f(\boldsymbol{x}) = \sum_{x_2} \sum_{x_3} f_1(x_1) f_2(x_2) f_3(x_1, x_2, x_3) \tag{2.70}$$

for the f from (2.69). Here, $\sum_{\sim x_1}$ denotes summation over all variables except x_1. Assuming that each variable is binary, direct computation of (2.70) for a particular value of x_1 requires 8 multiplications and 3 additions. However, using the distributive law we can rewrite (2.70) as follows:

$$f(x_1) = f_1(x_1) \sum_{x_2} f_2(x_2) \sum_{x_3} f_3(x_1, x_2, x_3). \tag{2.71}$$

Computing the expression in (2.71) requires only 3 multiplications and 3 additions, i.e., compared to (2.70) the number of operations is almost halved. The marginalization in (2.71) can equivalently be computed by performing message passing on the factor graph corresponding to (2.69). To this end, we write the marginal at variable node x_1 as the product of all incoming messages (cf. Figure 2.5), i.e., we have

$$f(x_1) = \mu_{f_1 \to x_1}(x_1) \mu_{f_3 \to x_1}(x_1). \tag{2.72}$$

These messages are given by[4]

$$\mu_{f_1 \to x_1}(x_1) = f_1(x_1), \tag{2.73a}$$

$$\mu_{f_3 \to x_1}(x_1) = \sum_{x_2} \mu_{x_2 \to f_3}(x_2) \sum_{x_3} f_3(x_1, x_2, x_3) \mu_{x_3 \to f_3}(x_3), \tag{2.73b}$$

where

$$\mu_{x_2 \to f_3}(x_2) = \mu_{f_2 \to x_2}(x_2) = f_2(x_2), \quad \text{and} \quad \mu_{x_3 \to f_3}(x_3) = 1. \tag{2.73c}$$

[4]We shall presently discuss a systematic way for the computation of the messages in (2.73).

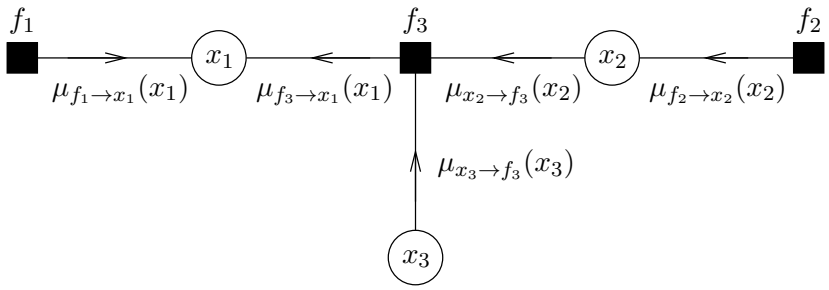


Figure 2.5: Message passing for the computation of the marginal $f(x_1) = \sum_{\sim x_1} f(\boldsymbol{x})$.

It is not hard to see that using (2.73) in (2.72) indeed yields (2.71). While for our toy example the message passing approach might seem rather artificial, it has proven extremely useful in signal processing and communications [68]. Application examples include channel estimation, Kalman filtering, iterative (turbo) detection, and channel decoding. In fact, many classical algorithms (like the BCJR algorithm [4], the Viterbi algorithm [100], etc.) can be viewed as specific instances of message passing on factor graphs.

We have seen above that in order to obtain a single marginal, we need to pass one message along every edge of the graph. The computation of more than one marginal follows the same principle as explained above. However, a key observation is that many of the intermediate messages from one marginalization can be reused for the computation of other marginals. Indeed, provided that the factor graph is a tree, passing two messages along every edge (one message in each direction) allows us to compute *all* marginals simultaneously. Therefore, computing all marginals on a tree requires just twice the computational complexity of computing a single marginal. The corresponding message passing scheme is known as the *sum-product algorithm* [54, Section II.C]. We will next summarize the sum-product algorithm for computing all marginals on a tree.

1. *Initialization:* Message passing is initialized at the leaf nodes of the tree. All vertices with degree 1 are leaf nodes, i.e., leaf nodes have exactly one neighbor. In case a factor node f_1 is a leaf node connected to variable node x_1, the initial message is $\mu_{f_1 \to x_1}(x_1) = f_1(x_1)$. For leaf variable nodes the initial message is $\mu_{x_1 \to f_1}(x_1) = 1$. The initialization at leaf nodes is depicted in Figure 2.6a.

2. *Message updates:* All internal vertices have at least two neighbors and can compute an outgoing message as soon as they have received a message from all but one of its neighbors. The update rule for a factor node f_1 with degree L and incoming messages $\mu_{x_l \to f_1}(x_l)$, $l = 2, \ldots, L$, is

$$\mu_{f_1 \to x_1}(x_1) = \sum_{\sim x_1} f_1(x_1, \ldots, x_L) \prod_{l=2}^{L} \mu_{x_l \to f_1}(x_l). \tag{2.74}$$

   Hence, the outgoing message $\mu_{f_1 \to x_1}(x_1)$ is the product of all incoming messages times

the local function f_1 marginalized with respect to all variables except x_1. For a variable node x_1 with degree J and incoming messages $\mu_{f_j \to x_1}(x_1)$, $j = 2, \ldots, J$, the update rule is

$$\mu_{x_1 \to f_1}(x_1) = \prod_{j=2}^{J} \mu_{f_j \to x_1}(x_1). \tag{2.75}$$

In this case, the outgoing message is simply the product of all incoming messages. The message update rules (2.74) and (2.75) are depicted in Figure 2.6b.

3. *Marginalization:* Computing the marginal $f(x_1)$ for a variable node x_1 with degree J requires incoming messages on all edges. Given the messages $\mu_{f_j \to x_1}$, $j = 1, \ldots, J$, the marginal $f(x_1)$ is

$$f(x_1) = \prod_{j=1}^{J} \mu_{f_j \to x_1}(x_1). \tag{2.76}$$

The marginalization operation in (2.76) is shown in Figure 2.6c. All marginals can be computed if the message updates (2.74) and (2.75) are performed until each variable node has received a message from all its neighbors. The sum-product algorithm thus terminates after a finite number of steps.

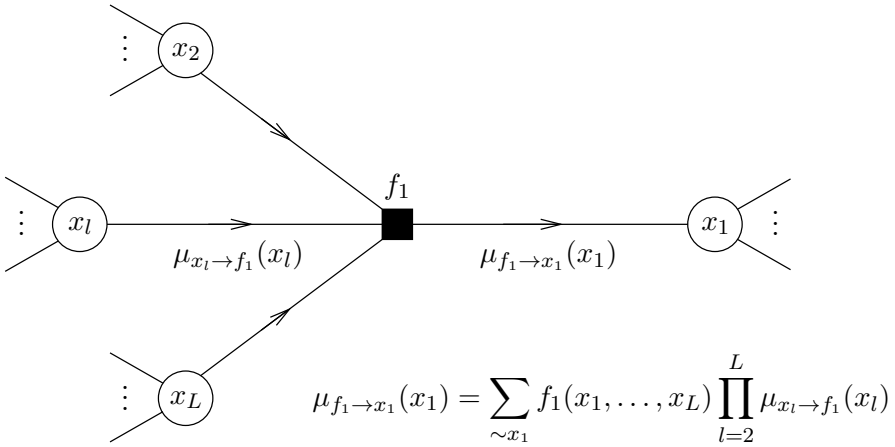
The order in which message are passed on the factor graph is specified by a so-called message passing *schedule* and typically there is a large number of possible schedules. In case the factor graph is a tree, the sum-product algorithm will always yield the exact marginals irrespective of the schedule. Therefore, we are free to choose any convenient schedule without changing the result.

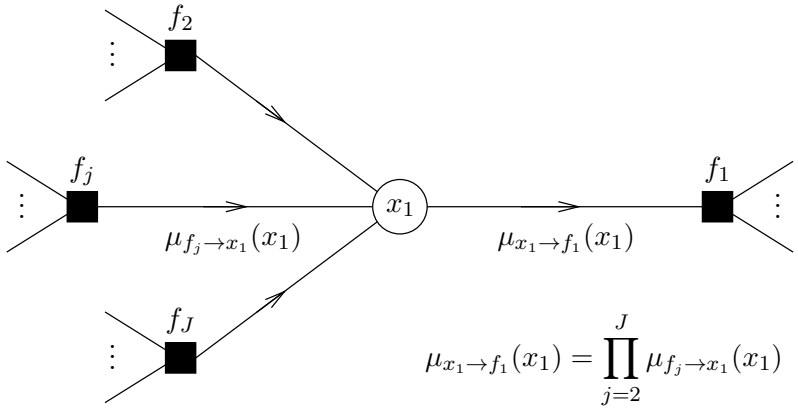
Unfortunately, this is no longer true if the factor graph contains cycles which is, e.g., the case for good channel codes. In this case, we can still run the sum-product algorithm with some modifications but we no longer obtain the exact marginals. Also, the messages need not converge and the result will generally depend on the employed schedule in a very intricate manner. Despite these problems, the sum-product algorithm performs astonishingly well in many applications where factor graphs contain cycles (e.g., in channel decoding).

The modifications we need to consider for the sum-product algorithm on graphs with cycles concern the initialization and the termination of the algorithm. A simple way to initialize message passing is to treat every vertex as if it were a leaf node in a tree. In this way, every node is able to pass a message along any edge after the initialization. During the message updates a new message simply replaces the previous message on that edge. Message passing is performed until some suitable stopping criterion is fulfilled. In the simplest case, the sum-product algorithm terminates after a fixed number of message updates.

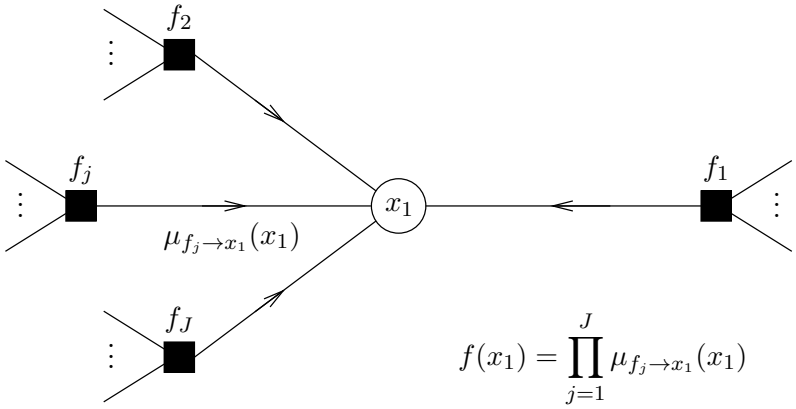
So far we have implicitly assumed that the variables take values from a finite set. This is motivated by our application of the sum-product algorithm in Chapter 6 where we perform message passing decoding for binary random variables. In the binary case, each message can be represented by a scalar value instead of a function. This fact is used in the next section to

(a) Initialization.





(b) Message updates.



(c) Marginalization.

Figure 2.6: Summary of the sum-product algorithm.

derive elegant formulations of channel decoding algorithms in terms of log-likelihood ratios (LLRs). The sum-product algorithm can also be used with continuous random variables. In this case sums have to be replaced by integrals in (2.74).

The key idea behind the sum-product algorithm is to exploit the distributive law. Marginals are computed efficiently because multiplication distributes over summation. For the sum-product algorithm it is required that the codomain of the global function f is a semiring with two operations "+" and "$\cdot$" [102, Section 3.6] that satisfy the distributive law

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c) \tag{2.77}$$

for all a, b, c in the codomain of f. In the "max-product" semiring we have the following distributive law for nonnegative real-valued quantities:

$$a \max\{b, c\} = \max\{ab, ac\}. \tag{2.78}$$

Similarly, for real-valued quantities "+" distributes over "min" yielding the "min-sum" semiring with the distributive law

$$a + \min\{b, c\} = \min\{a + b, a + c\}. \tag{2.79}$$

The above semirings allow us to formulate modified versions of the sum-product algorithm. From (2.78) we obtain the "max-product" algorithm which can equivalently be formulated in the logarithmic domain yielding the "max-sum" algorithm. These algorithms can be used to find maximum probability configurations in an efficient manner. An application example for max-product and max-sum is maximum likelihood (ML) sequence detection (e.g., the Viterbi algorithm). The "min-sum" algorithm can be derived using the distributive law in (2.79). This algorithm yields an efficient implementation of approximate MAP decoding of binary codes. Indeed, it can be shown that applying the max-log approximation $\log \sum_i e^{x_i} \approx \max_i x_i$ in the sum-product algorithm yields the min-sum algorithm.

2.6 Soft Information and Codes on Graphs

The concept of *soft information* is instrumental for any advanced receiver design. A common form of soft information in communication receivers are (approximations of) posterior probabilities of the transmitted data. For binary data, soft information is most conveniently expressed in terms of LLRs. In contrast to a hard decision, soft information additionally captures the reliability of a decision. It is well-known that soft information processing improves performance compared to hard decisions [36]. As an example, consider coded transmission of binary data over an additive white Gaussian noise (AWGN) channel. Figure 2.7 shows the bit error rate (BER) versus the signal-to-noise ratio (SNR) for hard-decision decoding (blue curve, '+' markers) and for soft-decision decoding (red curve, '$\times$' markers). We observe that

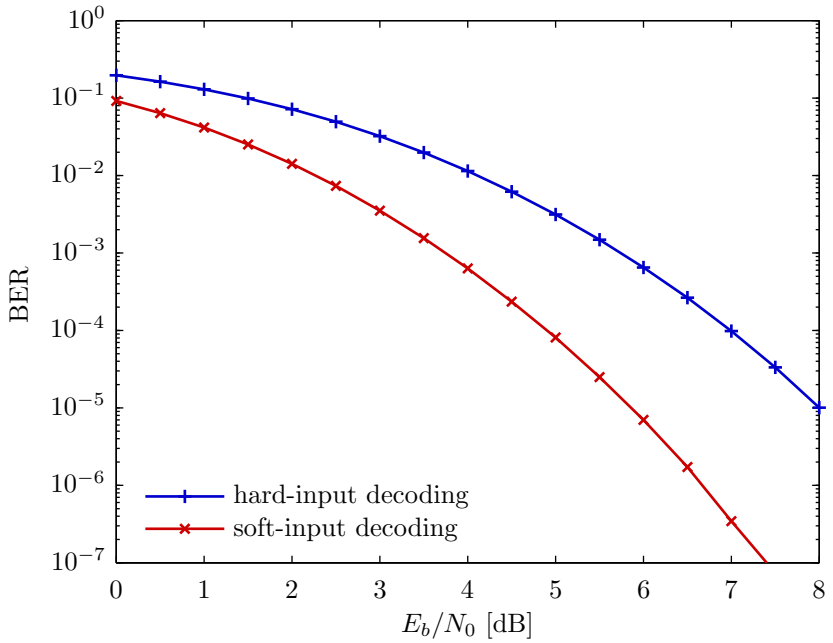


Figure 2.7: BER versus E_b/N_0 of coded transmission over an AWGN channel with hard-input and soft-input Viterbi decoding. The channel code is a $(7,5)_8$ convolutional code and the code bits are transmitted using binary phase-shift keying (BPSK).

soft-decision decoding yields a significant performance improvement. The key idea in modern receiver designs is to avoid hard decisions whenever possible and rather process reliability information instead. Iterative "turbo" receivers are based on exchanging soft information between receiver components. The turbo concept has revolutionized communication theory after the discovery of turbo codes [9] in 1993. Detailed treatises on coding theory are, e.g., given in [64, 87, 90].

2.6.1 Log-Likelihood Ratios

Throughout this thesis we focus on binary linear codes. Therefore, we next discuss soft information processing in the binary case formulated in terms of LLRs (which are sometimes also called "L-values"). Assume we have a binary[5] random variable $\mathsf{x} \in \{-1, 1\}$ with known prior probabilities $\mathbb{P}\{\mathsf{x} = -1\} = 1 - \mathbb{P}\{\mathsf{x} = 1\}$. Let x be transmitted over a channel with transition pdf $p(\boldsymbol{y}|x)$ yielding a received vector $\mathbf{y}$. At this point the channel is not assumed to be memoryless. Given an observation $\boldsymbol{y}$ at the channel output, the posterior LLR for x is given by[6]

$$L_{\mathsf{x}}(\boldsymbol{y}) = \log \frac{\mathbb{P}\{\mathsf{x}=1|\mathbf{y}=\boldsymbol{y}\}}{\mathbb{P}\{\mathsf{x}=-1|\mathbf{y}=\boldsymbol{y}\}}. \tag{2.80}$$

[5] Here, we use the alphabet $\{-1, 1\}$ instead of $\{0, 1\}$ for notational convenience.

[6] Some authors define LLRs as the negative value of (2.80). Our definition is convenient when we consider the LLR of a modulo-2 sum of independent binary random variables.

We note that $L_{\mathsf{x}}(\boldsymbol{y})$ is a sufficient statistic for x. The posterior probabilities can be recovered from $L_{\mathsf{x}}(\boldsymbol{y})$ as follows:

$$\mathbb{P}\{\mathsf{x}{=}x|\mathbf{y}{=}\boldsymbol{y}\} = \frac{1}{1+e^{-xL_{\mathsf{x}}(\boldsymbol{y})}}, \quad x \in \{-1,1\}. \tag{2.81}$$

Applying Bayes' rule to (2.80), we obtain

$$L_{\mathsf{x}}(\boldsymbol{y}) = \log\frac{p(\boldsymbol{y}|\mathsf{x}{=}1)\mathbb{P}\{\mathsf{x}{=}1\}}{p(\boldsymbol{y}|\mathsf{x}{=}-1)\mathbb{P}\{\mathsf{x}{=}-1\}} = \log\frac{p(\boldsymbol{y}|\mathsf{x}{=}1)}{p(\boldsymbol{y}|\mathsf{x}{=}-1)} + \log\frac{\mathbb{P}\{\mathsf{x}{=}1\}}{\mathbb{P}\{\mathsf{x}{=}-1\}} = L_{\mathsf{x}}^{c}(\boldsymbol{y}) + L_{\mathsf{x}}^{a}. \tag{2.82}$$

Hence, the posterior LLR $L_{\mathsf{x}}(\boldsymbol{y})$ can be written as the sum of the channel information

$$L_{\mathsf{x}}^{c}(\boldsymbol{y}) \triangleq \log\frac{p(\boldsymbol{y}|\mathsf{x}{=}1)}{p(\boldsymbol{y}|\mathsf{x}{=}-1)} \tag{2.83}$$

that depends on the observed channel output $\boldsymbol{y}$ and the channel model $p(\boldsymbol{y}|x)$, and the prior information

$$L_{\mathsf{x}}^{a} \triangleq \log\frac{\mathbb{P}\{\mathsf{x}{=}1\}}{\mathbb{P}\{\mathsf{x}{=}-1\}} \tag{2.84}$$

that depends on the prior probabilities of x. When $\mathbb{P}\{\mathsf{x}{=}1|\mathbf{y}{=}\boldsymbol{y}\} > \mathbb{P}\{\mathsf{x}{=}-1|\mathbf{y}{=}\boldsymbol{y}\}$ then we have $L_{\mathsf{x}}(\boldsymbol{y}) > 0$ and, conversely, $\mathbb{P}\{\mathsf{x}{=}1|\mathbf{y}{=}\boldsymbol{y}\} < \mathbb{P}\{\mathsf{x}{=}-1|\mathbf{y}{=}\boldsymbol{y}\}$ implies $L_{\mathsf{x}}(\boldsymbol{y}) < 0$. Hence, the sign of $L_{\mathsf{x}}(\boldsymbol{y})$ corresponds to the decision $\hat{x}(\boldsymbol{y})$ of a MAP detector, i.e., the hard decision corresponding to $L_{\mathsf{x}}(\boldsymbol{y})$ is $\hat{x}(\boldsymbol{y}) = \operatorname{sign}\big(L_{\mathsf{x}}(\boldsymbol{y})\big)$. The magnitude $|L_{\mathsf{x}}(\boldsymbol{y})|$ corresponds to the *reliability* of the associated hard decision. We are certain about the value of x if $|L_{\mathsf{x}}(\boldsymbol{y})| = \infty$ and, on the other hand, we know nothing about the value of x if $L_{\mathsf{x}}(\boldsymbol{y}) = 0$.

2.6.2 Extrinsic Information

We next turn our attention to soft information processing for channel codes. Consider two data bits $\mathsf{c}_1, \mathsf{c}_2 \in \{0,1\}$ which are channel-encoded by a single parity bit c_3. The coding rule is

$$\mathsf{c}_3 = \mathsf{c}_1 \oplus \mathsf{c}_2, \tag{2.85}$$

where "$\oplus$" denotes modulo-2 addition. We note that (2.85) is equivalent to $\mathsf{c}_3 = \mathbb{1}\{\mathsf{c}_1 \neq \mathsf{c}_2\}$ and to $\mathsf{c}_1 \oplus \mathsf{c}_2 \oplus \mathsf{c}_3 = 0$. The code (i.e., the set of codewords) defined by (2.85) is

$$\mathcal{C} = \{\boldsymbol{c} \in \{0,1\}^3 : c_1 \oplus c_2 \oplus u_3 = 0\}, \tag{2.86}$$

where $\boldsymbol{c} = (c_1\ c_2\ c_3)^{\mathrm{T}} \in \mathcal{C}$ denotes the vector of code bits. A code of the form (2.86) is called *single parity-check code*. We assume that the code bits are transmitted over a memoryless channel with transition pdf

$$p(\boldsymbol{y}|\boldsymbol{c}) = \prod_{k=1}^{3} p(y_k|c_k), \tag{2.87}$$

where $\boldsymbol{y} = (y_1\ y_2\ y_3)^{\mathrm{T}}$ denotes the vector of channel outputs. The posterior probability of c_k given $\boldsymbol{y}$ is given by

$$\mathbb{P}\{\mathsf{c}_k = b | \mathbf{y} = \boldsymbol{y}\} = \sum_{\boldsymbol{c} \in \mathcal{C}: c_k = b} \mathbb{P}\{\mathbf{c} = \boldsymbol{c} | \mathbf{y} = \boldsymbol{y}\}, \qquad b \in \{0, 1\}. \tag{2.88}$$

It is important to note that the posterior probability in (2.88) depends on the whole receive vector since the code constraint (2.85) connects the individual bits and therefore each receive value contains information about the bit c_k. Using Bayes' rule we rewrite (2.88) as follows:

$$\mathbb{P}\{\mathsf{c}_k = b | \mathbf{y} = \boldsymbol{y}\} = \sum_{\boldsymbol{c} \in \mathcal{C}: c_k = b} \frac{p(\boldsymbol{y}|\boldsymbol{c})\mathbb{P}\{\mathbf{c} = \boldsymbol{c}\}}{p(\boldsymbol{y})} = \frac{1}{p(\boldsymbol{y})} \sum_{\boldsymbol{c} \in \mathcal{C}: c_k = b} \prod_{l=1}^{3} p(y_l | \mathsf{c}_l = c_l)\mathbb{P}\{\mathsf{c}_l = c_l\}. \tag{2.89}$$

Here we have assumed that the prior probability $\mathbb{P}\{\mathbf{c} = \boldsymbol{c}\}$ factors into $\prod_{l=1}^{3} \mathbb{P}\{\mathsf{c}_l = c_l\}$. Writing out (2.89) for c_1 yields

$$\begin{aligned}\mathbb{P}\{\mathsf{c}_1 = b | \mathbf{y} = \boldsymbol{y}\} = \frac{1}{p(\boldsymbol{y})} p(y_1 | \mathsf{c}_1 = b)\mathbb{P}\{\mathsf{u}_1 = b\} \Big[& p(y_2 | \mathsf{c}_2 = 0)\mathbb{P}\{\mathsf{c}_2 = 0\} p(y_3 | \mathsf{c}_3 = b)\mathbb{P}\{\mathsf{u}_3 = b\} \\ & + p(y_2 | \mathsf{c}_2 = 1)\mathbb{P}\{\mathsf{c}_2 = 1\} p(y_3 | \mathsf{c}_3 = \overline{b})\mathbb{P}\{\mathsf{c}_3 = \overline{b}\} \Big],\end{aligned} \tag{2.90}$$

where $\overline{b} \triangleq b \oplus 1$.

Using (2.90) the posterior LLR for c_1 can be written as follows:

$$\begin{aligned} L_{\mathsf{c}_1}(\boldsymbol{y}) &= \log \frac{\mathbb{P}\{\mathsf{c}_k = 0 | \mathbf{y} = \boldsymbol{y}\}}{\mathbb{P}\{\mathsf{c}_k = 1 | \mathbf{y} = \boldsymbol{y}\}} && (2.91) \\ &= \log \frac{p(y_1 | \mathsf{c}_1 = 0)\mathbb{P}\{\mathsf{c}_1 = 0\}}{p(y_1 | \mathsf{c}_1 = 1)\mathbb{P}\{\mathsf{c}_1 = 1\}} \\ &\quad + \log \frac{\mathbb{P}\{\mathsf{c}_2 = 0 | \mathsf{y}_2 = y_2\}\mathbb{P}\{\mathsf{c}_3 = 0 | \mathsf{y}_3 = y_3\} + \mathbb{P}\{\mathsf{c}_2 = 1 | \mathsf{y}_2 = y_2\}\mathbb{P}\{\mathsf{c}_3 = 1 | \mathsf{y}_3 = y_3\}}{\mathbb{P}\{\mathsf{c}_2 = 0 | \mathsf{y}_2 = y_2\}\mathbb{P}\{\mathsf{c}_3 = 1 | \mathsf{y}_3 = y_3\} + \mathbb{P}\{\mathsf{c}_2 = 1 | \mathsf{y}_2 = y_2\}\mathbb{P}\{\mathsf{c}_3 = 0 | \mathsf{y}_3 = y_3\}} && (2.92) \\ &= L^c_{\mathsf{c}_1}(y_1) + L^a_{\mathsf{c}_1} + L^e_{\mathsf{c}_1}(\boldsymbol{y}_{\sim 1}), && (2.93) \end{aligned}$$

where $\boldsymbol{y}_{\sim 1}$ denotes the vector that is obtained by removing the first element of $\boldsymbol{y}$, i.e., $\boldsymbol{y}_{\sim 1} = (y_2\ y_3)^{\mathrm{T}}$. The term $L^e_{\mathsf{c}_1}(\boldsymbol{y}_{\sim 1})$ in (2.93) is called *extrinsic information*, i.e., the knowledge about c_1 that *other* bits of a codeword contribute. The notion of extrinsic information is fundamental for advanced iterative receivers. To simplify the expression in (2.92) we rewrite the extrinsic LLR in terms of the posterior LLRs for c_2 and c_3 as

$$L^e_{\mathsf{c}_1}(\boldsymbol{y}_{\sim 1}) = \log \frac{1 + \dfrac{\mathbb{P}\{\mathsf{c}_2 = 0 | \mathsf{y}_2 = y_2\}\mathbb{P}\{\mathsf{c}_3 = 0 | \mathsf{y}_3 = y_3\}}{\mathbb{P}\{\mathsf{c}_2 = 1 | \mathsf{y}_2 = y_2\}\mathbb{P}\{\mathsf{c}_3 = 1 | \mathsf{y}_3 = y_3\}}}{\dfrac{\mathbb{P}\{\mathsf{c}_2 = 0 | \mathsf{y}_2 = y_2\}}{\mathbb{P}\{\mathsf{c}_2 = 1 | \mathsf{y}_2 = y_2\}} + \dfrac{\mathbb{P}\{\mathsf{c}_3 = 0 | \mathsf{y}_3 = y_3\}}{\mathbb{P}\{\mathsf{c}_3 = 1 | \mathsf{y}_3 = y_3\}}} = \log \frac{1 + \exp\big(L_{\mathsf{c}_2}(y_2) + L_{\mathsf{c}_3}(y_3)\big)}{\exp\big(L_{\mathsf{c}_2}(y_2)\big) + \exp\big(L_{\mathsf{c}_3}(y_3)\big)}. \tag{2.94}$$

2.6.3 The Boxplus Operator

To simplify notation and to write (2.94) more compactly, we introduce the "boxplus" operator $\boxplus$ which is defined as follows [40]:

$$a \boxplus b \triangleq \frac{1+e^{a+b}}{e^a+e^b} = 2\,\mathrm{atanh}\big(\tanh(a/2)\tanh(b/2)\big). \tag{2.95}$$

This allows us to write (2.94) as $L^e_{\mathsf{c}_1}(\boldsymbol{y}_{\sim 1}) = L_{\mathsf{c}_2}(y_2) \boxplus L_{\mathsf{c}_3}(y_3)$. It is not hard to see that for two independent bits c_1 and c_2, the boxplus operator (2.95) yields the LLR of $\mathsf{c}_1 \oplus \mathsf{c}_2$, i.e., we have $L_{\mathsf{c}_1} \boxplus L_{\mathsf{c}_2} = L_{\mathsf{c}_1 \oplus \mathsf{c}_2}$. The boxplus operator has the following properties:

- *Closure*: For all $a, b \in \mathbb{R} \Rightarrow a \boxplus b \in \mathbb{R}$.
- *Associativity*: For all $a, b, c \in \mathbb{R} \Rightarrow (a \boxplus b) \boxplus c = a \boxplus (b \boxplus c)$.
- *Commutativity*: For all $a, b \in \mathbb{R} \Rightarrow a \boxplus b = b \boxplus a$.
- *Identity element*: For any $a \in \mathbb{R} \Rightarrow a \boxplus \infty = a$.
- We have $0 \boxplus a = 0$ and $\pm\infty \boxplus a = \pm a$.
- An inverse element does not exist since $|a \boxplus b| \leq \min\{|a|, |b|\}$.

The first four of the properties above imply that the algebraic structure of $(\mathbb{R} \cup \{\pm\infty\}, \boxplus)$ is a commutative monoid with ∞ as the identity element.

In the above development we have restricted ourselves to a single parity-check code with three bits. With the boxplus notation we can easily extend (2.93) to an arbitrary number of bits. Specifically, let $\mathcal{C} = \{\boldsymbol{c} \in \{0,1\}^n : c_1 \oplus \cdots \oplus c_n = 0\}$ be a single parity-check code of length n. The posterior LLR for the bit c_k then is

$$L_{\mathsf{c}_k}(\boldsymbol{y}) = L^c_{\mathsf{c}_k}(y_k) + L^a_{\mathsf{c}_k} + \sum_{\substack{j=1 \\ j \neq k}}^{n} \boxplus\, L_{\mathsf{c}_j}(y_j), \tag{2.96}$$

where we have used the shorthand notation

$$\sum_{j=1}^{n} \boxplus\, a_j \triangleq a_1 \boxplus \cdots \boxplus a_n = 2\,\mathrm{atanh} \prod_{j=1}^{n} \tanh(a_j/2). \tag{2.97}$$

The last term in the sum of (2.96) is again the extrinsic information for c_k. We note that (2.96) is the MAP-optimal decoder for a single parity-check code, i.e., $\hat{c}_k = \mathrm{sign}\big(L_{\mathsf{c}_k}(\boldsymbol{y})\big)$ minimizes the bit error probability.

Since the computation of the boxplus sum in (2.96) involves the evaluation of transcendental functions, approximations of the boxplus operator are of interest. To this end we first rewrite (2.95) as follows:

$$a \boxplus b = 2\,\mathrm{sign}(a)\,\mathrm{sign}(b)\,\mathrm{atanh}\big(\tanh(|a|/2)\tanh(|b|/2)\big). \tag{2.98}$$

Noting that $\operatorname{atanh}\big(\tanh(|a|/2)\tanh(|b|/2)\big) \leq \operatorname{atanh}\big(\tanh(\min\{|a|,|b|\}/2)\big)$ yields the simple approximation

$$a \mathbin{\widetilde{\boxplus}} b \triangleq \operatorname{sign}(a)\operatorname{sign}(b)\min\{|a|,|b|\} \approx a \boxplus b \tag{2.99}$$

which overestimates the reliability of $a \boxplus b$, i.e., $\min\{|a|,|b|\} \geq |a \boxplus b|$. The approximation in (2.99) is good if the magnitudes of a and b are far apart. In the worst case, i.e., when $a = b$, we have $\Delta = a \mathbin{\widetilde{\boxplus}} b - a \boxplus b \leq \log(2)$, where $\Delta \approx \log(2)$ is a good approximation if $|a| = |b| \geq 3$. Another approximation of the boxplus operator can be obtained by applying the max-log approximation to (2.95). We have

$$a \boxplus b = \max\{0, a+b\} - \max\{a,b\} - \log\frac{1+e^{-|a-b|}}{1+e^{-|a+b|}} \tag{2.100}$$

$$\approx \max\{0, a+b\} - \max\{a,b\} = a \mathbin{\widetilde{\boxplus}} b. \tag{2.101}$$

We note that (2.101) is equal to (2.99). Applying the max-log approximation to (2.95) is useful since it provides an additive correction term which can be used to refine the approximate boxplus $\widetilde{\boxplus}$. By storing a few values of the last term in (2.100), a very accurate approximation of the boxplus operation can be implemented using a lookup table. Similarly, we can a find multiplicative correction by rewriting (2.95) as [106]

$$a \boxplus b = a \mathbin{\widetilde{\boxplus}} b \left(1 - \frac{1}{\min\{|a|,|b|\}} \log\frac{1+e^{-\big||a|-|b|\big|}}{1+e^{-\big||a|+|b|\big|}}\right). \tag{2.102}$$

The multiplicative correction in (2.102) is attractive because the correction factor is bounded between 0 and 1. Using (2.99), a low-complexity implementation of the MAP-optimal decoder (2.96) is given by

$$L_{\mathsf{c}_k}(\boldsymbol{y}) = L^c_{\mathsf{c}_k}(y_k) + L^a_{\mathsf{c}_k} + \Big[\prod_{j=1, j\neq k}^{n} \operatorname{sign}\big(L_{\mathsf{c}_j}(y_j)\big)\Big] \min_{j=1,\ldots,n; j\neq k} |L_{\mathsf{c}_j}(y_j)|. \tag{2.103}$$

The decoding rule in (2.103) is widely known as the *min-sum decoder* which also results from message passing using the min-sum algorithm as discussed in Section 2.5. The decoders (2.96) and (2.103) are examples of *soft-input soft-output* decoders. We note that soft-input soft-output decoders are the main building blocks of iterative receivers.

2.6.4 Linear Block Codes

A binary code $\mathcal{C}$ is *linear* if and only if for any $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{C}$ we have $\boldsymbol{c}_1 \oplus \boldsymbol{c}_2 \in \mathcal{C}$. Linear codes are conveniently described using matrices. Specifically, a linear block code $\mathcal{C}$ of *dimension* K and *blocklength* N can be described by an $(N-K) \times N$ *parity-check matrix* $\boldsymbol{H}$. The code $\mathcal{C}$ is equal to the nullspace of $\boldsymbol{H}$, i.e., we have

$$\mathcal{C} = \{\boldsymbol{c} \in \{0,1\}^N \mid \boldsymbol{H}\boldsymbol{c} = \boldsymbol{0}\}. \tag{2.104}$$

Here, all matrix-vector products are in the finite field GF(2). A detailed introduction to the mathematics of finite fields can be found, e.g., in [63]. Each row of $\boldsymbol{H}$ corresponds to one *parity-check equation*. Equivalently, the same linear code $\mathcal{C}$ can be described by an $N \times K$ *generator matrix* $\boldsymbol{G}$ which satisfies[7]

$$\boldsymbol{H}\boldsymbol{G} = \boldsymbol{0}. \tag{2.105}$$

Therefore, we have

$$\mathcal{C} = \left\{ \boldsymbol{c} \in \{0,1\}^N \,|\, \boldsymbol{c} = \boldsymbol{G}\boldsymbol{u},\ \forall \boldsymbol{u} \in \{0,1\}^K \right\}. \tag{2.106}$$

Hence, the code $\mathcal{C}$ is the column space of $\boldsymbol{G}$. If (2.105) is fulfilled, then (2.104) and (2.106) describe the same code, since for any $\boldsymbol{u}$ we then have $\boldsymbol{H}\boldsymbol{G}\boldsymbol{u} = \boldsymbol{H}\boldsymbol{c} = \boldsymbol{0}$. The *rate* R of a linear block code $\mathcal{C}$ is defined as

$$R \triangleq \frac{K}{N} \in (0,1]. \tag{2.107}$$

The code rate R characterizes the amount of redundancy introduced by the code. A low-rate code (i.e., R close to zero) introduces a lot of redundancy and can be expected to have strong error correction capabilities. Conversely, a high-rate code (i.e., $R \approx 1$) has little redundancy and therefore its error correction capabilities are weak.

As an example, let us consider a binary linear code $\mathcal{C}$ described by the following parity-check matrix:

$$\boldsymbol{H} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.108}$$

This code has rate $R = 1/2$ with $K = 4$ and $N = 8$. Hence, it consists of $2^4 = 16$ codewords. An equivalent description of the code $\mathcal{C}$ is in terms of the following generator matrix:

$$\boldsymbol{G} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}. \tag{2.109}$$

From (2.109) it can be seen that the code in our example is a *systematic* code, i.e., the codewords are of the form

$$\boldsymbol{c} = \boldsymbol{G}\boldsymbol{u} = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix}. \tag{2.110}$$

[7]Here, the generator matrix is a tall matrix because we write $\boldsymbol{u}$ and $\boldsymbol{c}$ as column vectors (which is in contrast to some textbooks).

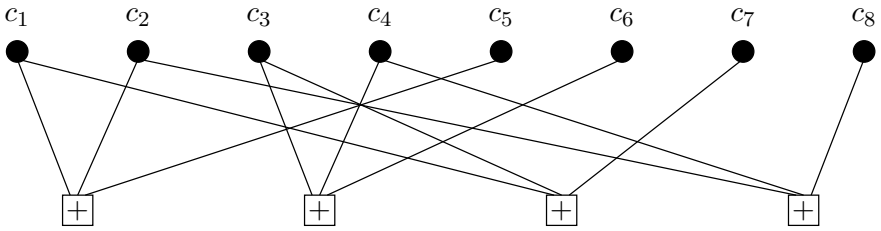


Figure 2.8: Tanner graph corresponding to the parity-check matrix in (2.108).

Here, the information bits $\boldsymbol{u}$ and the parity bits $\boldsymbol{p}$ appear separately in the codeword $\boldsymbol{c}$ which is advantageous in some applications.

Given a noisy codeword $\boldsymbol{y}$, observed at the output of a channel with transition pdf $p(\boldsymbol{y}|\boldsymbol{c})$, an ML decoder finds

$$\hat{\boldsymbol{c}}(\boldsymbol{y}) = \underset{\boldsymbol{c}\in\mathcal{C}}{\arg\max}\, p(\boldsymbol{y}|\boldsymbol{c}). \tag{2.111}$$

The ML decoder computes the codeword $\hat{\boldsymbol{c}} \in \mathcal{C}$ that most likely caused the observation $\boldsymbol{y}$ under the channel model $p(\boldsymbol{y}|\boldsymbol{c})$. Similarly, a MAP decoder computes

$$\hat{c}_l(\boldsymbol{y}) = \underset{c_l\in\{0,1\}}{\arg\max}\, p(c_l|\boldsymbol{y}) = \underset{c_l\in\{0,1\}}{\arg\max} \sum_{\boldsymbol{c}_{\sim l}} p(\boldsymbol{c}|\boldsymbol{y}), \quad l = 1, \ldots, N. \tag{2.112}$$

The MAP decoder maximizes the posterior probability for each bit separately. Therefore, the MAP decoder minimizes the bit error probability although $\tilde{\boldsymbol{c}} = (\hat{c}_1 \ \cdots \ \hat{c}_N)^{\mathrm{T}}$ may not be a valid codeword (which is in contrast to the ML decoder). While the decoders in (2.111) and (2.112) may be feasible for the code in the above example, their computational complexity scales exponentially with the blocklength N. The maximization in (2.111) and the marginalization in (2.112) have to take all 2^{NR} codewords into account which is prohibitively complex for codes of practical interest. Therefore, low-complexity decoders with near-optimal performance are of great interest for decoding channel codes. To this end, we next discuss a graphical representation of linear block codes.

2.6.5 Iterative Decoding

Linear block codes can be represented by a *Tanner graph*. Tanner graphs are bipartite graphs with *variable nodes* (one per code bit) and *check nodes* (one per parity-check equation). The ith check node is connected to the jth variable node by an edge if $\boldsymbol{H}_{i,j} = 1$. Figure 2.8 depicts the Tanner graph corresponding to the parity-check matrix in (2.108). A linear block code is composed of $N - K$ component codes, where each component code is a single parity-check code. Therefore, a simple (but suboptimal) decoding strategy is to optimally decode each component code and exchange extrinsic information between the individual components in an iterative manner. Since Tanner graphs can be viewed as a particular form of factor graphs, such an iterative decoder can be derived using the message passing rules of Section 2.5.

We next describe the process of iterative decoding, and the update rules for variables nodes and check nodes in terms of LLRs. First, the variable nodes are initialized with the channel information, i.e., we have[8]

$$L_{\mathsf{c}_l,0} = L^c_{\mathsf{c}_l}(y_l), \quad l = 1, \ldots, N. \tag{2.113}$$

The check nodes then compute extrinsic LLRs as in (2.96). Specifically, in the kth iteration a check node connected to the J variable nodes $\mathsf{c}_1, \ldots, \mathsf{c}_J$ computes the following boxplus sums:

$$L^e_{\mathsf{c}_l,k} = \sum_{\substack{j=1 \\ j\neq l}}^{J} \boxplus\, L_{\mathsf{c}_j,k}, \quad l = 1, \ldots, J. \tag{2.114}$$

These extrinsic LLRs are then used as additional prior information at the variable nodes. Hence, a variable node c_l connected to, say, M check nodes updates its LLR according to

$$L_{\mathsf{c}_l,k+1} = L_{\mathsf{c}_l,k} + \sum_{m=1}^{M} L^{e,m}_{\mathsf{c}_l,k}. \tag{2.115}$$

Here, $L_{\mathsf{c}_l,k}$ denotes the previous LLR value and $L^{e,m}_{\mathsf{c}_l,k}$ is the extrinsic LLR received from the mth check node in the kth iteration. In the next iteration, the updated LLRs of (2.115) are used to compute new extrinsic LLRs using (2.114). Decoding is stopped if all parity-check constraints are fulfilled or if a given number of iterations (or message updates) has been performed. The hard decisions correspond to the sign of the variable node LLRs in (2.115).

The sum of extrinsic LLRs in (2.115) is due to the assumption that the underlying bits are statistically independent. If the Tanner graph contains cycles, this independence assumption is violated and therefore iterative decoding is suboptimal. We are free to update the nodes in any order, but the choice of the schedule may influence the convergence and the performance of the decoder. A common approach is the so-called *flooding schedule* which updates all messages from the check nodes to the variable nodes in one time instant and in the next time instant all messages from the variables nodes to the check nodes are updated.

The message update at the check nodes involves the computation of a boxplus sum which is rather complicated compared to the update at the variable nodes. The min-sum decoder avoids this problem by approximating the boxplus operation as in (2.99). Hence, the check node update of the min-sum decoder is

$$L^e_{\mathsf{c}_l,k} = \Big[\prod_{\substack{j=1 \\ j\neq l}}^{J} \operatorname{sign}\big(L_{\mathsf{c}_j,k}\big)\Big] \min_{\substack{j=1,\ldots,J \\ j\neq l}} |L_{\mathsf{c}_j,k}|, \quad l = 1, \ldots, J. \tag{2.116}$$

We note that the update rule (2.116) is a suboptimal decoder for the single parity-check component codes which is in contrast to (2.114). The simplification due to the boxplus

[8] The additional index on the left-hand side of (2.113) corresponds to the iteration number.

Table 2.1: Initial values of the variable nodes.

$L_{\mathsf{c}_1,0}$	$L_{\mathsf{c}_2,0}$	$L_{\mathsf{c}_3,0}$	$L_{\mathsf{c}_4,0}$	$L_{\mathsf{c}_5,0}$	$L_{\mathsf{c}_6,0}$	$L_{\mathsf{c}_7,0}$	$L_{\mathsf{c}_8,0}$
0.5	1.5	3.5	1.0	1.0	-1.5	2.0	-2.5

Table 2.2: Values of the variable nodes after the first iteration.

$L_{\mathsf{c}_1,1}$	$L_{\mathsf{c}_2,1}$	$L_{\mathsf{c}_3,1}$	$L_{\mathsf{c}_4,1}$	$L_{\mathsf{c}_5,1}$	$L_{\mathsf{c}_6,1}$	$L_{\mathsf{c}_7,1}$	$L_{\mathsf{c}_8,1}$
3.5	1.0	3.0	-2.0	1.5	-0.5	2.5	-1.5

approximation (2.99) causes a performance penalty which can be reduced by appropriately down-scaling the extrinsic LLRs [61].

We next give an example to illustrate graph-based iterative decoding on the Tanner graph in Figure 2.8. For simplicity, we assume that a min-sum decoder with flooding schedule is used. First, we initialize the variable node LLRs with the channel information $L^c_{\mathsf{c}_l}(y_l)$. The initial values $L_{\mathsf{c}_l,0}$, $l = 1, \ldots, 8$, are given in Table 2.1. If the initial variable node LLRs satisfy all parity-check equations, we may stop at this point. However, this is not the case in our example. The hard decision of the LLRs in Table 2.1 corresponds to the bits $(0\ 0\ 0\ 0\ 0\ 1\ 0\ 1)^{\mathrm{T}}$. These bits do not form a codeword since the second and the fourth parity-check equation are not satisfied. The nearest valid codeword in terms of Hamming distance is $(0\ 0\ 0\ 1\ 0\ 1\ 0\ 1)^{\mathrm{T}}$, i.e., the channel has most probably corrupted the fourth bit of the transmitted codeword.

To start the iterative decoding process, the variable nodes pass their values to the check nodes. At the check nodes, extrinsic LLRs are computed according to (2.116). These extrinsic LLRs are then passed to the variable nodes (cf. Figure 2.9) where they are used as new prior information. The updated variable node LLRs (cf. Table 2.2) are given by the sum of the values in Table 2.1 and the extrinsic LLRs. This completes the first iteration of the decoding process. In this iteration, the sign of $L_{\mathsf{c}_4,1}$ has changed compared to $L_{\mathsf{c}_4,0}$. Thus, all parity-check equations are now satisfied. At this point we may stop decoding since a valid codeword has been found. It is important to note that the codeword found by the decoder need not be equal to the transmitted codeword.

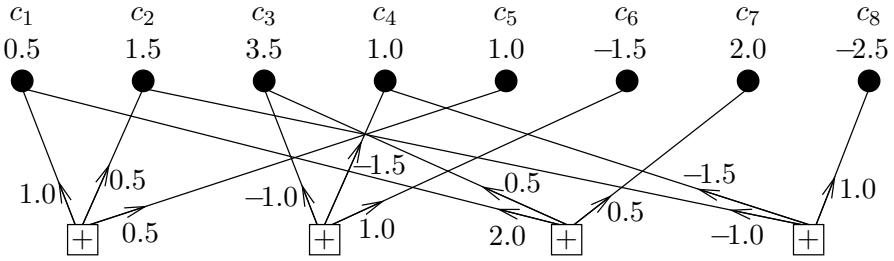


Figure 2.9: Extrinsic LLRs in the first iteration.

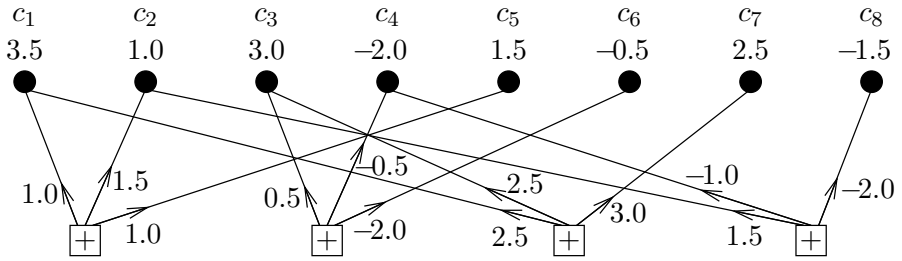


Figure 2.10: Extrinsic LLRs in the second iteration.

Table 2.3: Values of the variable nodes after the second iteration.

$L_{\mathsf{c}_1,2}$	$L_{\mathsf{c}_2,2}$	$L_{\mathsf{c}_3,2}$	$L_{\mathsf{c}_4,2}$	$L_{\mathsf{c}_5,2}$	$L_{\mathsf{c}_6,2}$	$L_{\mathsf{c}_7,2}$	$L_{\mathsf{c}_8,2}$
7.0	4.0	6.0	−3.5	2.5	−2.5	5.5	−3.5

In some cases the decoder may perform a fixed number of iterations without performing a parity check after each iteration. In our example, the second iteration proceeds in the same manner as the first iteration. The extrinsic LLRs are shown in Figure 2.10 and the updated variable node LLRs are given in Table 2.3. None of the values at the variable nodes have changed their sign in the second iteration, but all LLRs have increased their magnitude[9]. This is an important point because if we kept on iterating, the magnitude of some LLRs may grow without bound. This shows that iterative decoding is suboptimal and the messages being passed on the graph are no longer exact LLRs since their magnitude does not correspond to the reliability of the associated hard decision. From an implementation point of view this means that the messages have to be "clipped" to avoid numerical problems. The performance impact of LLR clipping can be mitigated by appropriately up-scaling the clipped values [91].

2.6.6 Low-Density Parity-Check Codes

Low-density parity-check (LDPC) codes were first introduced by Gallager in the early 1960's [29]. At that time it had not been recognized that LDPC codes could closely approach channel capacity for sufficiently large blocklengths. Gallager's work on LDPC codes has been largely ignored until these codes were rediscovered in the 1990's [69] after the invention of turbo codes [9].

An LDPC code is a linear block code given by the nullspace of a parity-check matrix $\boldsymbol{H}$ with a low density of nonzero entries. While there is no precise definition of the term "low-density", typical LDPC codes have parity-check matrices with less than 0.1 % nonzero entries. The sparsity property of $\boldsymbol{H}$ allows us to efficiently decode LDPC codes using iterative algorithms with near-optimal performance. In this context, the iterative message passing decoder described in the previous subsection is often called *belief propagation* (BP) decoder.

[9]This is also the case if a sum-product decoder is used instead of the min-sum decoder.

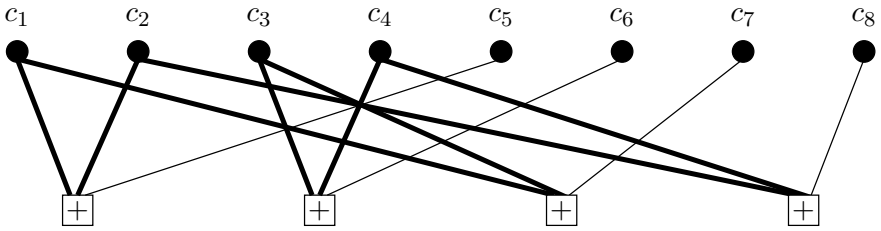


Figure 2.11: A cycle of length 8 (indicated by bold lines) in the Tanner graph of Figure 2.8.

As discussed above, the BP decoder is suboptimal for codes whose Tanner graph contains cycles. However, it can be shown that codes without cycles cannot perform well since they suffer from many low-weight codewords which yields a high probability of error [87, Section 2.6]. Hence, cycles are critical for the performance of LDPC codes under BP decoding. We note that few cycles are bad for the code but good for the BP decoder and, conversely, many cycles are good for the code but bad for the BP decoder. It turns out that short cycles deteriorate the performance of the BP algorithm and should therefore be avoided by design of the parity-check matrix. Figure 2.11 shows a cycle of length 8 in the Tanner graph of our example from the previous subsection. In this example there are no shorter cycles. The length of the shortest cycle in a graph is called its *girth*. Hence, the girth of the Tanner graph in Figure 2.8 is 8.

The analysis of LDPC codes is usually performed in terms of *ensembles* which is easier than characterizing a particular LDPC code. A code ensemble is a class of codes with common properties. Density evolution [86] allows us to analyze the average behavior of LDPC ensembles under BP decoding in the limit of large blocklength. This average analysis is useful because it can be shown that as $N \to \infty$, almost all codes of an ensemble behave alike. An example is the ensemble of (λ, ρ)-*regular* LDPC codes. Here, each variable node has degree λ and each check node has degree ρ, i.e., each code bit participates in λ check equations and each check equation is the modulo-2 sum of ρ code bits. Equivalently, each row of the parity-check matrix of a (λ, ρ)-regular LDPC code has ρ nonzero entries and each column has λ nonzero entries. A generalization of the regular ensemble is the ensemble of $\big(\lambda(x), \rho(x)\big)$-*irregular* LDPC codes with variable node degree distribution

$$\lambda(x) = \sum_i \lambda_i x^i \tag{2.117}$$

and check node degree distribution

$$\rho(x) = \sum_i \rho_i x^i. \tag{2.118}$$

Here, λ_i is the fraction of variable nodes of degree i and, similarly, ρ_i is the fraction of check nodes of degree i.

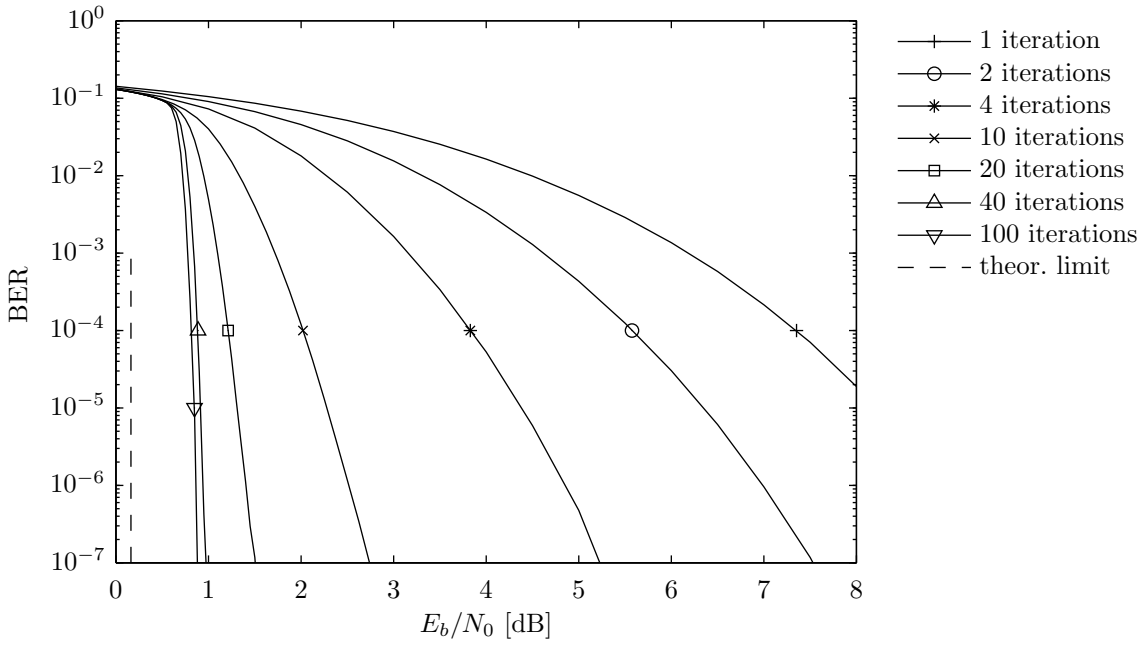
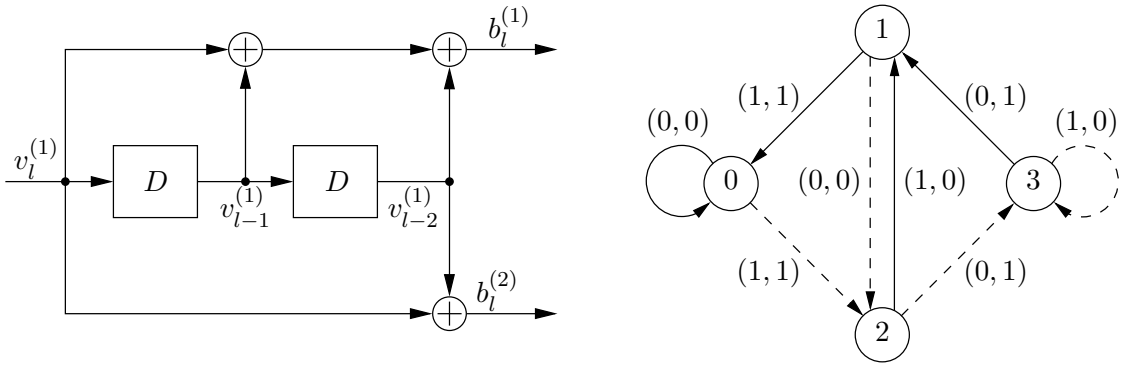


Figure 2.12: BER performance of the rate-1/2 DVB-S2 LDPC code (blocklength 64800 bits). The code bits are transmitted using BPSK over an AWGN channel. A soft-input BP decoder has been used to obtain these results.

Results for a particular code, i.e., for a specific member of an ensemble, can be obtained using Monte Carlo simulations. Figure 2.12 depicts a BER versus SNR plot of the rate-1/2 DVB-S2 code with a blocklength of 64800 bits [23]. We observe that up to approximately 40 decoder iterations large SNR gains can be achieved. A further increase in the number of iterations yields only a marginal performance improvement. Moreover, we can see that this code performs within a few tenths of a dB of the theoretical limit at a BER of 10^{-7}. In [18], a rate-1/2 irregular LDPC code has been constructed whose density evolution threshold is 0.0045 dB away from the Shannon limit. For a blocklength of 10^7 bits, this code performs within 0.04 dB of the theoretical limit at a BER of 10^{-6}.

2.6.7 Convolutional Codes and the BCJR Algorithm

Convolutional Codes. In contrast to block codes, convolutional codes are not restricted to a fixed blocklength. Convolutional codes map a (possibly infinite) stream of information bits to a stream of code bits. Encoders for convolutional codes process data in small blocks called *frames*. The sequence of information bits is split up into data frames $\boldsymbol{v}_l = (v_l^{(1)} \cdots v_l^{(k)})^{\mathrm{T}}$ of length k. Each data frame is mapped to a length-n ($n > k$) code frame $\boldsymbol{b}_l = (b_l^{(1)} \cdots b_l^{(n)})^{\mathrm{T}}$ by the encoder. The resulting code rate is $R = k/n$. Encoding of convolutional codes is not memoryless; the encoder output at frame time l depends not only on its input but also on the *state* S_l of the encoder. We can write the code frames and the evolution of the encoder



(a) Encoder implementation.

(b) State transition diagram.

Figure 2.13: Rate-1/2 convolutional encoder with (a) shift register implementation and (b) state transition diagram. In (b), solid lines correspond to $v_l^{(1)} = 0$, dashed lines correspond to $v_l^{(1)} = 1$, and the edges are labeled by the encoder output $(b_l^{(1)}, b_l^{(2)})$.

state as follows ($l = 0, 1, \ldots$):

$$\boldsymbol{b}_l = \chi(\boldsymbol{v}_l, S_l), \tag{2.119}$$

$$S_{l+1} = \psi(\boldsymbol{v}_l, S_l). \tag{2.120}$$

Here, χ denotes the output function, ψ denotes the state transition function, and the initial state S_0 is known. We note that due to (2.120), the sequence of states is a Markov chain.

An efficient way to implement encoders for convolutional codes, i.e., the equations (2.119) and (2.120), is through linear shift register circuits. Figure 2.13a shows an encoder implementation for a rate-1/2 convolutional code with $k = 1$ and $n = 2$. The encoder state corresponds to the content of the shift register. In our example, we enumerate the states by the decimal equivalent of $(v_{l-1}^{(1)}, v_{l-2}^{(1)})$, where $v_{l-1}^{(1)}$ is the most significant bit. A state transition diagram corresponding to the encoder in Figure 2.13a is shown in Figure 2.13b. The edges indicate transitions between states, where a solid line corresponds to $v_l^{(1)} = 0$ and a dashed line corresponds to $v_l^{(1)} = 1$. Furthermore, each edge is labeled by the encoder output $(b_l^{(1)}, b_l^{(2)})$ which results from that particular state transition and input.

Encoders can be conveniently described using generator polynomials. Generator polynomials are usually given in octal notation and they specify which inputs and which shift register taps are used in the computation of the code bits. In our example, the first code bit $b_l^{(1)}$ is the modulo-2 sum of the input and both shift register taps. These connections correspond to the binary string 111_2 (with the input being the most significant bit) which is 7_8 in octal representation. Similarly, the second code bit $b_l^{(2)}$ is the modulo-2 sum of the input with the last shift register tap, corresponding to 101_2, i.e., 5_8 in octal representation. Therefore, we write the generator polynomials of the encoder in Figure 2.13a as $(7, 5)_8$.

The encoder in Figure 2.13a is a feedforward shift register circuit and yields a nonsystematic code. Systematic codes can be generated by shift register circuits with feedback.

The *encoder memory* (more precisely, the *frame memory order*) is the minimum number of data frames required to force the encoder state from an arbitrary state to the initial state S_0. Equivalently, the encoder memory is the maximum length of all shift registers in the encoder. The number of states is (at most) two to the power of the number of shift register taps in the encoder. We note that k and n are small integers (typically $k = 1$) and therefore convolutional codes are rather limited regarding their rate. To achieve higher code rates *puncturing* is commonly employed.

As mentioned earlier, convolutional codes are not restricted to a particular blocklength. While this is a favorable property, it is sometimes required to encode data blocks of a fixed length. Convolutional codes can be used in those cases by transforming them to linear block codes using *truncation* or *termination*. The codeword of a truncated convolutional code consists of all code bits produced by the data. Hence, the encoder ends in some arbitrary state which is reset to the initial state S_0 before the next block is encoded. In the case of termination, the data is padded with additional frames to force the encoder state to the initial state S_0. The codeword includes the code bits that are due to the termination and the next block can be encoded immediately since the encoder is already in state S_0. We note that truncation does not change the code rate, but termination leads to a small rate loss which is often negligible if the blocklength is sufficiently large. In contrast to block codes, the performance of convolutional codes mainly depends on the encoder memory instead of the blocklength. While convolutional codes perform reasonably well also for shorter blocklengths, they do not approach channel capacity. However, convolutional codes are the building blocks of turbo codes which indeed perform close to the Shannon limit.

Convolutional codes are usually decoded using the Viterbi algorithm [100], which efficiently performs ML decoding. Sequential decoding techniques like the Fano algorithm [24] and the stack algorithm [49] are well suited for decoding convolutional codes with large encoder memory. Modern receiver concepts often require channel decoders which output soft information. To this end, the soft-output Viterbi algorithm [39], soft-output stack algorithms, e.g., [15], the so-called LISS algorithm [37], and the BCJR algorithm [4] can be used. Of these algorithms, only the BCJR algorithm is a MAP-optimal soft-output decoder; the other algorithms are suboptimal. In the following we describe the BCJR algorithm in more detail.

The BCJR algorithm. Let $\boldsymbol{u} = (u_1 \cdots u_K)^{\mathrm{T}}$ be a length-K vector of information bits which is encoded and transmitted over a memoryless channel. The length-N codeword corresponding to $\boldsymbol{u}$ is $\boldsymbol{c} = (c_1 \cdots c_N)^{\mathrm{T}} \in \mathcal{C}$. The channel output is used to compute the LLRs $\boldsymbol{L}^c = (L^c_{\mathsf{c}_1} \cdots L^c_{\mathsf{c}_N})^{\mathrm{T}}$, i.e., the channel information for the code bits. Additionally, prior information about the data bits is given by the LLRs $\boldsymbol{L}^a = (L^a_{\mathsf{u}_1} \cdots L^a_{\mathsf{u}_K})^{\mathrm{T}}$. Using $\boldsymbol{L}^c$ and $\boldsymbol{L}^a$, we want to efficiently compute the posterior distributions

$$p(c_l|\boldsymbol{L}^c, \boldsymbol{L}^a) = \sum_{\boldsymbol{c}_{\sim l}} p(\boldsymbol{c}|\boldsymbol{L}^c, \boldsymbol{L}^a), \quad l = 1, \ldots, N, \tag{2.121}$$

$$p(u_l|\boldsymbol{L}^c, \boldsymbol{L}^a) = \sum_{\boldsymbol{u}_{\sim l}} p(\boldsymbol{u}|\boldsymbol{L}^c, \boldsymbol{L}^a), \quad l = 1, \ldots, K. \tag{2.122}$$

From (2.121) and (2.122) we can easily compute the posterior LLRs for the code bits and the information bits, respectively. The BCJR algorithm provides an efficient way to perform the marginalizations in (2.121) and (2.122).

Let us consider a rate-k/n convolutional code $\mathcal{C}$ with M states and let $S_l \in \mathcal{S}$ denote the encoder state at (frame) time l, where $\mathcal{S} = \{0, \ldots, M-1\}$ is the set of states. Due to the frame structure of the encoder, the sequence of information bits $\boldsymbol{u}$ is split info T data frames $\boldsymbol{v}_l = \left(v_l^{(1)} \cdots v_l^{(k)}\right)^{\mathrm{T}}$, $l = 1, \ldots, T$, of length $k = K/T$. Similarly, the codeword $\boldsymbol{c}$ is split into T code frames $\boldsymbol{b}_l = \left(b_l^{(1)} \cdots b_l^{(n)}\right)^{\mathrm{T}}$, $l = 1, \ldots, T$, of length $n = N/T$. To describe the BCJR algorithm in detail, we first introduce the notion of a trellis. A trellis is a graph which represents the evolution of the encoder state over time. At each time instant we have M vertices which represent the encoder states. An edge connects the vertices corresponding to $S_l = m'$ and $S_{l+1} = m$ if the encoder is such that the state transition $m' \to m$ is possible. We denote the set of state tuples which correspond to possible state transitions by $\mathcal{T} \subseteq \mathcal{S} \times \mathcal{S}$, i.e., $(m', m) \in \mathcal{T}$ implies that the transition $m' \to m$ is possible. Since the encoder input is binary, each vertex has 2^k outgoing edges. Every state transition is associated to a particular input frame and a particular output frame.

A simple way to construct one section of the trellis is using the state transition diagram of the encoder. Figure 2.14a shows a trellis section for the encoder depicted in Figure 2.13a. The trellis of convolutional codes is time-invariant[10] and thus the entire trellis is simply a concatenation of T trellis sections. Since the initial encoder state S_0 is known, the edges leaving other states are removed in the first section of the trellis. For a terminated convolutional code, edges are also removed from the last few sections of the trellis to ensure termination in the final state S_T. Without loss of generality we may assume that $S_0 = 0$ and, in the case of a terminated code, $S_T = 0$. A complete trellis description of the encoder in our example is depicted in Figure 2.14b.

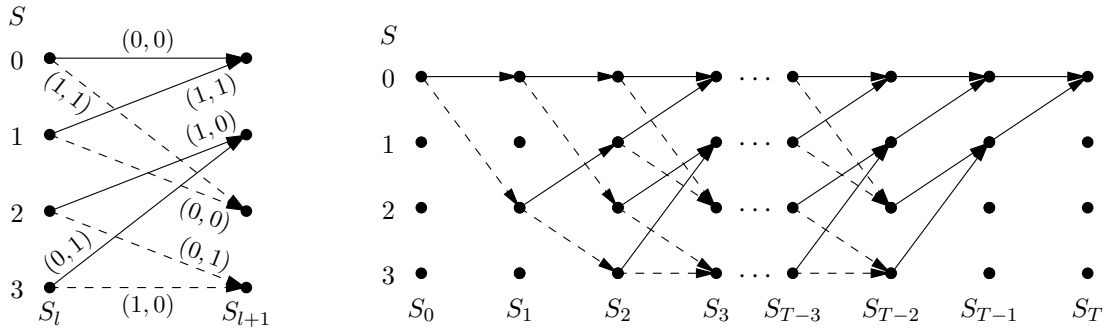
The BCJR algorithm allows us to write the marginalizations in (2.121) and (2.122) as follows:

$$\mathbb{P}\{\mathsf{b}_l^{(j)} = c|\boldsymbol{L}^c, \boldsymbol{L}^a\} = \sum_{(m',m)\in\mathcal{A}_c^{(j)}} \alpha_{l-1}(m')\gamma_l(m', m)\beta_l(m), \quad j = 1, \ldots, n, \quad l = 1, \ldots, T, \tag{2.123}$$

$$\mathbb{P}\{\mathsf{v}_l^{(j)} = u|\boldsymbol{L}^c, \boldsymbol{L}^a\} = \sum_{(m',m)\in\mathcal{B}_u^{(j)}} \alpha_{l-1}(m')\gamma_l(m', m)\beta_l(m), \quad j = 1, \ldots, k, \quad l = 1, \ldots, T. \tag{2.124}$$

Here, $\mathcal{A}_c^{(j)}$ denotes the set of state transitions $m' \to m$ such that the jth bit of the code frame

[10] The BCJR algorithm can also be applied to block codes with time-varying trellises. However, the construction of trellises for block codes (discussed, e.g., in [64, Chapter 9]) goes beyond the scope of this discussion.



(a) Trellis section. (b) Complete trellis for T frames.

Figure 2.14: Trellis description of the encoder in Figure 2.13a. Solid lines correspond to $v_l^{(1)} = 0$ and dashed lines correspond to $v_l^{(1)} = 1$. The edges in (a) are labeled in the same way as in the state transition diagram (cf. Figure 2.13b).

equals c. Similarly, the set $\mathcal{B}_u^{(j)}$ contains all state transitions $m' \to m$ where the jth bit of the data frame is equal to u. We note that the sets $\mathcal{A}_c^{(j)}$ and $\mathcal{B}_u^{(j)}$ are time-invariant, i.e., they do not depend on l. The role of $\gamma_l(m', m)$ is discussed later in more detail. The key insight which makes the BCJR algorithm efficient is that the quantities α_l and β_l can be computed recursively. In the *forward recursion*, α_l is computed using α_{l-1} as follows:

$$\alpha_l(m) = \sum_{m'=0}^{M-1} \alpha_{l-1}(m')\gamma_l(m', m), \quad m = 0, \ldots, M-1, \quad l = 1, \ldots, T. \tag{2.125}$$

The *backward recursion* computes β_l using β_{l+1} as follows:

$$\beta_l(m) = \sum_{m'=0}^{M-1} \beta_{l+1}(m')\gamma_{l+1}(m, m'), \quad m = 0, \ldots, M-1, \quad l = T-1, \ldots, 1. \tag{2.126}$$

The initializations for these recursions are $\alpha_0(m) = \mathbb{1}\{m = 0\}$, and $\beta_T(m) = \mathbb{1}\{m = 0\}$ (in the case of a terminated code) or $\beta_T(m) = 1/M$ (for a truncated code).

The α's, β's, and γ's in the BCJR algorithm are actually probabilities. However, in our case we compute the state transition probabilities $\gamma(m', m)$ such that they are not normalized. Specifically, we have

$$\gamma_l(m', m) = \mathbb{1}\{(m', m) \in \mathcal{T}\} \prod_{i=1}^{n} \frac{\exp\left(-c_{m',m}^{(i)} L^c_{\mathsf{b}_l^{(i)}}\right)}{1 + \exp\left(-L^c_{\mathsf{b}_l^{(i)}}\right)} \prod_{j=1}^{k} \frac{\exp\left(-u_{m',m}^{(j)} L^a_{\mathsf{v}_l^{(j)}}\right)}{1 + \exp\left(-L^a_{\mathsf{v}_l^{(j)}}\right)}, \quad l = 1, \ldots, T, \tag{2.127}$$

where $c_{m',m}^{(i)}$ is the ith output bit of the encoder in the state transition $m' \to m$, $u_{m',m}^{(j)}$ is the jth input bit of the encoder in the state transition $m' \to m$, and $L^c_{\mathsf{b}_l^{(i)}}$, $L^a_{\mathsf{v}_l^{(j)}}$ are the ith channel LLR and the jth prior LLR in the lth code and data frame, respectively. Since $\gamma_l(m', m)$ in (2.127) is not normalized, we need to normalize the α's and β's after each step.

It is important to note that due to the trellis structure described by the set $\mathcal{T}$, we have $\gamma_l(m', m) = 0$ for some tuples (m', m) and therefore the sums in (2.125) and (2.126) actually contain less than M terms. A compact description of the BCJR algorithm for soft-input soft-output channel decoding is given in Algorithm 2.1.

Some remarks regarding the BCJR algorithm are in order. The initialization of Algorithm 2.1 assumes a terminated code. For a truncated code the initialization of β_T must be changed to $\beta_T(m) = 1/M$, $m = 0, \ldots, M-1$. We have chosen the initial encoder state S_0 and the terminal encoder state S_T to be zero. However, this choice is arbitrary and can be changed in the initialization of the BCJR algorithm. It is rather simple to extend Algorithm 2.1 to punctured convolutional codes. Indeed, it is sufficient to insert channel LLRs with value zero at the positions of the punctured code bits and then run the BCJR algorithm on the trellis of the unpunctured code. To avoid numerical problems it may be required to clip large LLR magnitudes to a maximum value of, say, 20. For systematic encoders, the marginalization for the posterior LLRs for the information bits can be skipped since the codeword contains all information bits. For encoders with a feedforward shift register structure, the marginalization for the jth information bit in the lth frame simplifies to $\sum_{m \in \tilde{\mathcal{B}}_u^{(j)}} \alpha_l(m)\beta_l(m)$, where $\tilde{\mathcal{B}}_u^{(j)} = \{m : (m', m) \in \mathcal{B}_u^{(j)}, \forall m' \in \mathcal{S}\}$. Furthermore, it is important to note that Algorithm 2.1 does not assume any particular channel model. The channel model enters through the LLRs $\boldsymbol{L}_c$ which are computed before the BCJR algorithm is executed.

A reformulation of Algorithm 2.1 in the logarithmic domain is sometimes convenient because we work with LLRs. The BCJR algorithm in the logarithmic domain is also known as the log-MAP decoder. By applying the max-log approximation to the log-MAP decoder we arrive at the max-log-MAP decoder which reduces complexity since it avoids the evaluation of transcendental functions. A further complexity reduction can be achieved by retaining only large values of α_l and β_l in each step. These simplifications are known as the M-BCJR (keeps the M largest values) and the T-BCJR (keeps all values above a certain threshold) [28]. Furthermore, windowed decoding can be used to reduce memory requirements for codes with large blocklength and enables hardware implementation of the BCJR algorithm [7].

We note that the BCJR algorithm can be re-derived in a factor graph framework using the sum-product algorithm. The min-sum algorithm yields the max-log-MAP version of the BCJR decoder. The BCJR algorithm is a variant of the forward-backward algorithm which is used in machine learning to infer the posterior distribution of hidden state variables in hidden Markov models (HMMs) given a sequence of observations. Similarly, the Baum-Welch algorithm [5] uses the forward-backward procedure to infer the model parameters of HMMs with applications in, e.g., speech recognition and modeling of genomic sequences.

2.6.8 Turbo Codes

In 1948, Shannon proved that the channel capacity can be achieved using random codes when the blocklength tends to infinity [94]. Although a lot of research regarding the design

Algorithm 2.1 *Soft-input soft-output BCJR decoding algorithm.*

Input: $\boldsymbol{L}_a$, $\boldsymbol{L}_c$, T, trellis structure $(\{\mathcal{A}_c^{(i)}\}_{i=1}^n, \{c_{m',m}^{(i)}\}_{i=1}^n, \{\mathcal{B}_u^{(j)}\}_{j=1}^k, \{u_{m',m}^{(j)}\}_{j=1}^k, \mathcal{S}, \mathcal{T})$
Initialization: $\alpha_0(m) \leftarrow \mathbb{1}\{m{=}0\}$, $\beta_T(m) \leftarrow \mathbb{1}\{m{=}0\}$, $m = 0, \ldots, M-1$

 # compute transition probabilities
for $l = 1, \ldots, T$ **do**
  $\gamma_l(m', m) \leftarrow \mathbb{1}\{(m', m) \in \mathcal{T}\} \prod_{i=1}^{n} \frac{\exp\big(-c_{m',m}^{(i)} L^c_{\mathsf{b}_l^{(i)}}\big)}{1+\exp\big(-L^c_{\mathsf{b}_l^{(i)}}\big)} \prod_{j=1}^{k} \frac{\exp\big(-u_{m',m}^{(j)} L^a_{\mathsf{v}_l^{(j)}}\big)}{1+\exp\big(-L^a_{\mathsf{v}_l^{(j)}}\big)}, \quad \forall (m', m)$
end for

 # forward recursion
for $l = 1, \ldots, T$ **do**
  $\tilde{\alpha}_l(m) \leftarrow \sum_{m'=0}^{M-1} \alpha_{l-1}(m')\gamma_l(m', m), \quad m = 0, \ldots, M-1$
  $\alpha_l(m) \leftarrow \frac{\tilde{\alpha}_l(m)}{\sum_{m'=0}^{M-1} \tilde{\alpha}_l(m')}, \quad m = 0, \ldots, M-1$
end for

 # backward recursion
for $l = T-1, \ldots, 1$ **do**
  $\tilde{\beta}_l(m) \leftarrow \sum_{m'=0}^{M-1} \beta_{l+1}(m')\gamma_{l+1}(m, m'), \quad m = 0, \ldots, M-1$
  $\beta_l(m) \leftarrow \frac{\tilde{\beta}_l(m)}{\sum_{m'=0}^{M-1} \tilde{\beta}_l(m')}, \quad m = 0, \ldots, M-1$
end for

 # marginalization
for $l = 1, \ldots, T$ **do**
  for $i = 1, \ldots, n$ **do**
    $p_0 \leftarrow \sum_{(m',m)\in\mathcal{A}_0^{(i)}} \alpha_{l-1}(m')\gamma_l(m', m)\beta_l(m)$
    $p_1 \leftarrow \sum_{(m',m)\in\mathcal{A}_1^{(i)}} \alpha_{l-1}(m')\gamma_l(m', m)\beta_l(m)$
    $L_{\mathsf{c}_{(l-1)n+i}} \leftarrow \log p_0/p_1$
  end for
  for $j = 1, \ldots, k$ **do**
    $p_0 \leftarrow \sum_{(m',m)\in\mathcal{B}_0^{(j)}} \alpha_{l-1}(m')\gamma_l(m', m)\beta_l(m)$
    $p_1 \leftarrow \sum_{(m',m)\in\mathcal{B}_1^{(j)}} \alpha_{l-1}(m')\gamma_l(m', m)\beta_l(m)$
    $L_{\mathsf{u}_{(l-1)k+j}} \leftarrow \log p_0/p_1$
  end for
end for

Output: Posterior LLRs $L_{\mathsf{c}_{(l-1)n+i}}$, $L_{\mathsf{u}_{(l-1)k+j}}$, $l = 1, \ldots, T$, $i = 1, \ldots, n$, $j = 1, \ldots, k$, for code bits $\mathsf{c}_1, \ldots, \mathsf{c}_N$ and information bits $\mathsf{u}_1, \ldots, \mathsf{u}_K$

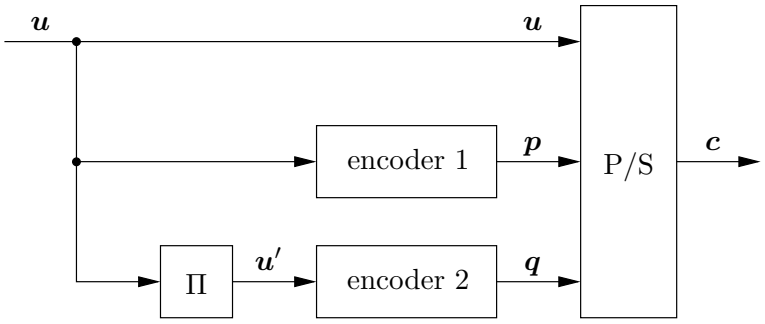


Figure 2.15: Block diagram of a turbo encoder with two constituent codes.

of good channel codes has been conducted, the discovery of the first practical codes which closely approach the Shannon limit took until 1993 [9]. These codes are called *turbo codes* and they mark the beginning of the modern era in coding theory. The fundamental idea of turbo codes is a code design that produces random-like codes with enough structure to enable efficient decoding. Turbo codes consist of two or more simple constituent codes along with pseudorandom interleavers. The constituent codes are concatenated in a parallel or serial manner. In principle, any code could be used as constituent code. However, we consider the usual case of parallel concatenated convolutional codes in the following.

Encoding of turbo codes is systematic and each constituent encoder produces parity bits for the data or an interleaved version of it. The constituent encoders are systematic encoders and the systematic bits are punctured from their output. Each codeword consists of the systematic part, i.e., the data itself, and the parity bits of the constituent encoders. Usually the constituent encoders are identical, but this need not be the case. Figure 2.15 shows the block diagram of a turbo encoder with two parallel concatenated encoders. Here, the first encoder produces parity bits for the data $\boldsymbol{u}$ and the second encoder computes parity bits for the interleaved data $\boldsymbol{u}' = \Pi(\boldsymbol{u})$, where Π denotes the interleaver. The input of the first encoder need not be interleaved and therefore a turbo encoder with M constituent codes will usually have $M - 1$ different interleavers. To achieve good performance the encoders need to have a recursive structure with feedback and the interleaver depth, i.e., the blocklength, needs to be rather large.

The interleaving is of fundamental importance for the performance of turbo codes. There are several (partly equivalent) ways to explain why interleaving is vital for the performance turbo codes.

- Interleaving allows us to create random-like codes. This is important since capacity can asymptotically be achieved using random coding arguments.

- A vast number of states is needed for a trellis representation of a good turbo code due to the interleaving. In general, a larger number of states yields a better performance of the code.

- As a consequence of interleaving, turbo codes are time-varying which is necessary to achieve *spectral thinning*, i.e., to reduce the multiplicities of low-weight codewords. Codes with fewer low-weight codewords feature a lower probability of error.

- Pseudorandom interleaving avoids short cycles in the factor graph representation of turbo codes. This is important for the performance of turbo codes when iterative decoding techniques are employed.

Optimal decoding of turbo codes, e.g., using the Viterbi algorithm or the BCJR algorithm, is infeasible due to the large number of encoder states. Therefore, suboptimal iterative decoding techniques are used in practice. The idea behind efficient decoding of turbo codes is similar to iterative decoding of block codes: optimally decode each constituent code and iteratively exchange extrinsic information between the individual component decoders. Figure 2.16 shows the block diagram of a turbo decoder corresponding to the encoder in Figure 2.15. The BCJR algorithm is used to perform MAP-optimal soft-input soft-output decoding of each constituent code. The first decoder operates on the LLRs $\boldsymbol{L}_{\mathbf{u}}^{c}$ and $\boldsymbol{L}_{\mathbf{p}}^{c}$ which stem from the channel observation. Initially, there is no prior information about $\mathbf{u}$, i.e., $\boldsymbol{L}_{\mathbf{u}}^{a} = \mathbf{0}$. Subtracting the prior LLRs $\boldsymbol{L}_{\mathbf{u}}^{a}$ from the posterior LLRs $\boldsymbol{L}_{\mathbf{u}}$ at the output of the first decoder yields extrinsic LLRs $\boldsymbol{L}_{\mathbf{u}}^{e}$. In the next step, these extrinsic LLRs are interleaved and used as new prior LLRs for the second decoder, i.e., $\boldsymbol{L}_{\mathbf{u}'}^{a} = \Pi(\boldsymbol{L}_{\mathbf{u}}^{e})$. The second decoder operates on the LLRs $\boldsymbol{L}_{\mathbf{u}'}^{c} = \Pi(\boldsymbol{L}_{\mathbf{u}}^{c})$ and $\boldsymbol{L}_{\mathbf{q}}^{c}$ from the channel observation together with the prior information $\boldsymbol{L}_{\mathbf{u}'}^{a}$. Extrinsic information from the second decoder is then deinterleaved (denoted by Π^{-1} in Figure 2.16) and is again used as new prior information for the first decoder. In this manner, the component decoders exchange extrinsic information for a certain number of iterations. Eventually, a hard decision is taken at the output of the first decoder[11] to obtain the decoded data $\hat{\boldsymbol{u}}$. Depending on the complexity and delay constraints, the number of turbo decoder iterations is usually between 2 and 10. The block-based processing depicted in Figure 2.16 corresponds to a particular schedule of a message passing decoder on the turbo code's factor graph.

Figure 2.17 shows the BER versus SNR performance of a rate-1/2 turbo code with an interleaver size and data blocklength of 2^{17} bits. The two constituent codes are equal with generator polynomials $(37, 21)_8$. We observe that the BER performance improves substantially in the first few iterations. The additional performance gain per iteration diminishes as the iteration count increases. Furthermore, we observe that the performance of this code is less than 1 dB away from the theoretical limit at a BER of 10^{-5}. The considered turbo code outperforms a rate-1/2 convolutional code with generator polynomials $(37, 21)_8$ after 2 iterations for BER values of practical interest. In Figure 2.17 we can see that turbo codes suffer from an error floor which causes the BER curve to flatten out for values below $\sim 10^{-5}$. Therefore, turbo codes may not be suitable for applications that operate at very low BERs.

[11] Alternatively, the hard decision can also be taken at the output of the second decoder yielding $\hat{\boldsymbol{u}}'$. Deinterleaving $\hat{\boldsymbol{u}}'$ yields the decoded data $\hat{\boldsymbol{u}} = \Pi^{-1}(\hat{\boldsymbol{u}}')$.

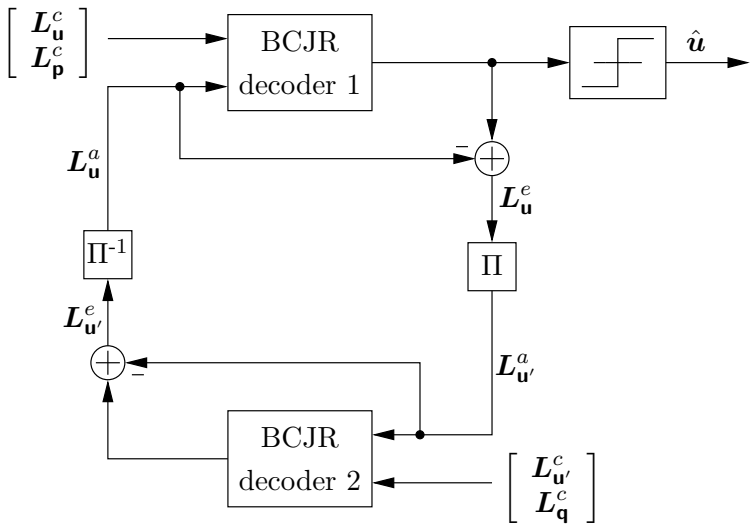


Figure 2.16: Block diagram of an iterative turbo decoder.

The error floor is caused by the weight distribution of turbo codes and in particular by their relatively poor minimum distance.

We have restricted our discussion of turbo codes to parallel concatenated convolutional codes with two constituent encoders. The extension to more than two constituent encoders is rather straightforward. For a discussion of turbo codes with serial concatenation of the constituent codes see, e.g., [6]. The idea of turbo processing has been extended to many applications beyond pure channel coding. Examples include turbo equalization [98], iterative detection for bit-interleaved coded modulation [62], and turbo source-channel coding [33]. Finally, we note that extrinsic information transfer charts [95] can be used to analyze and optimize turbo schemes.

2.7 The Information Bottleneck Method

In [97], Tishby et al. have introduced a novel method for data compression using the notion of *relevance through another variable*. The idea of the IB method is to compress the data $\mathsf{y} \in \mathcal{Y}$ such that its compressed version $\mathsf{z} \in \mathcal{Z}$ contains as much information as possible about the *relevance variable* $\mathsf{x} \in \mathcal{X}$, subject to a constraint on the compression rate. Figuratively speaking, the compression variable z constitutes a "bottleneck" through which the information that y provides about x is squeezed, hence the name "information bottleneck". In the IB setting, the joint distribution $p(x, y)$ is known and z depends only on y through a probabilistic mapping $p(z|y)$. Hence, the random variables x, y, and z form the Markov chain $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z}$. Furthermore, we assume that x, y, and z are discrete random variables, i.e., the cardinality of the sets $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ is finite.

The advantage of the IB framework over data compression in the rate-distortion (RD) setting is that the IB method avoids the choice of a distortion measure. In RD theory,

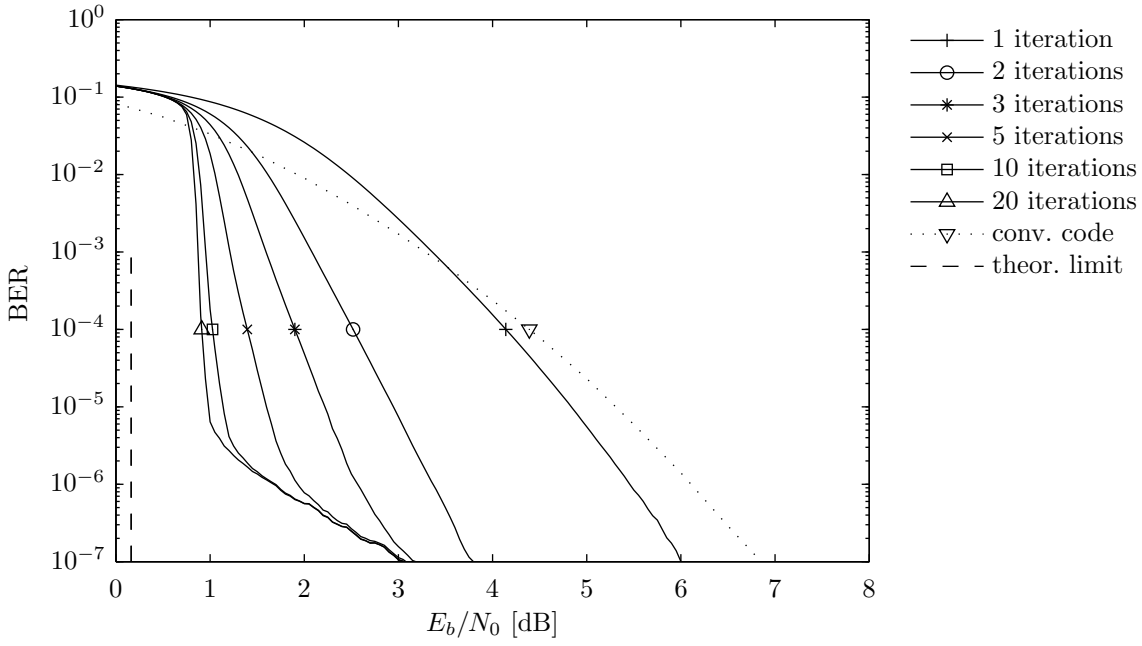


Figure 2.17: BER performance of a rate-1/2 turbo code with a blocklength of 2^{17} bits. The code bits are transmitted using BPSK over an AWGN channel. The turbo decoder in Figure 2.16 has been used to obtain these results.

a distortion measure has to be specified in advance and, in turn, the distortion measure determines which features of the data are relevant. This is problematic because there is no systematic way to find a suitable distortion measure for a given problem. Therefore, the distortion measure is often chosen in favor of mathematical tractability instead of perceptual meaningfulness [35, Section 2.4]. In contrast, the relevant features of the data are determined directly by the choice of the relevance variable x in the IB setting and, hence, the distortion measure *emerges* from the joint distribution $p(x, y)$.

The IB method finds the optimal compression mapping $p(z|y)$ as the solution of the variational problem

$$\min_{p(z|y)} I(\mathsf{y};\mathsf{z}) - \beta I(\mathsf{x};\mathsf{z}), \tag{2.128}$$

where the Lagrange parameter $\beta > 0$ controls the trade-off between the compression rate $I(\mathsf{y};\mathsf{z})$ and the relevant information $I(\mathsf{x};\mathsf{z})$. Large β yields little compression and thus much relevant information is preserved. Conversely, small β preserves little relevant information and entails strong compression. We note that in contrast to the RD problem, (2.128) is a non-convex problem and therefore cannot be solved using standard interior point methods [12, Chapter 11]. However, an implicit solution for the optimal assignment $p(z|y)$ is given by [97, Theorem 4]

$$p(z|y) = \frac{p(z)}{\psi(y,\beta)} \exp[-\beta D(p(x|y)\|p(x|z))]\,, \tag{2.129}$$

where $\psi(y,\beta)$ is the partition function, i.e., a normalization such that $p(z|y)$ is a valid probability distribution for each y. The expression in (2.129) is only an implicit solution of (2.128) since $p(z)$ and $p(x|z)$ depend on $p(z|y)$. We have

$$p(z) = \sum_{y\in\mathcal{Y}} p(y)p(z|y), \tag{2.130}$$

$$p(x|z) = \frac{1}{p(z)} \sum_{y\in\mathcal{Y}} p(x,y)p(z|y), \tag{2.131}$$

where (2.131) is due to the Markovity of $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z}$.

We next rewrite the relevant information $I(\mathsf{x};\mathsf{z})$. To this end, we note that (cf. (2.31))

$$I(\mathsf{x};\mathsf{y},\mathsf{z}) = I(\mathsf{x};\mathsf{z}) + I(\mathsf{x};\mathsf{y}|\mathsf{z}) = I(\mathsf{x};\mathsf{y}) + \underbrace{I(\mathsf{x};\mathsf{z}|\mathsf{y})}_{=0}. \tag{2.132}$$

Using (2.132), we can write the relevant information as

$$I(\mathsf{x};\mathsf{z}) = I(\mathsf{x};\mathsf{y}) - I(\mathsf{x};\mathsf{y}|\mathsf{z}). \tag{2.133}$$

Since the first term on the right-hand side of (2.133) does not depend on $p(z|y)$, we continue to rewrite $I(\mathsf{x};\mathsf{y}|\mathsf{z})$ as follows:

$$I(\mathsf{x};\mathsf{y}|\mathsf{z}) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \sum_{z\in\mathcal{Z}} p(x,y,z) \log \frac{p(x,y|z)}{p(x|z)p(y|z)} \tag{2.134}$$

$$= \sum_{y\in\mathcal{Y}} \sum_{z\in\mathcal{Z}} p(y,z) \sum_{x\in\mathcal{X}} p(x|y) \log \frac{p(x|y)}{p(x|z)} \tag{2.135}$$

$$= \mathbb{E}\big\{D\big(p(x|\mathsf{y})\|p(x|\mathsf{z})\big)\big\} = \mathbb{E}\{d(\mathsf{y},\mathsf{z})\}, \tag{2.136}$$

where we have defined

$$d(y,z) \triangleq D\big(p(x|y)\|p(x|z)\big). \tag{2.137}$$

Hence, we can equivalently write (2.128) as

$$\min_{p(z|y)} I(\mathsf{y};\mathsf{z}) + \beta\mathbb{E}\{d(\mathsf{y},\mathsf{z})\}. \tag{2.138}$$

The equivalence between (2.128) and (2.138) shows that the constraint on the relevant information is equivalent to a constraint on the expected relative entropy in (2.136). We note that (2.138) has the form of a RD problem with Lagrange parameter β. Therefore, it is natural to view $d(y,z)$ as the correct distortion measure in the IB setting. It is important to note that the distortion measure $d(y,z)$ depends on $p(z|y)$. This is in contrast to RD theory where the distortion measure is fixed *a priori*.

It can be shown that the IB equations (2.129)-(2.131) are satisfied simultaneously at the minima of the *free energy* $\mathbb{E}\{-\log\psi(\mathsf{y},\beta)\}$. The minimization of the free energy can

Algorithm 2.2 *Iterative IB algorithm.*

Input: $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, $p(x,y)$, $\beta > 0$, $\varepsilon > 0$, $M \in \mathbb{N}$

Initialization: $\eta \leftarrow \infty$, $\overline{d}^{(0)} \leftarrow \infty$, $m \leftarrow 1$, randomly choose $p^{(0)}(z|y)$

1: **while** $\eta \geq \varepsilon$ **and** $m \leq M$ **do**

2: $\quad p(z) \leftarrow \sum_{y \in \mathcal{Y}} p^{(m-1)}(z|y)p(y)$

3: $\quad p(x|z) \leftarrow \frac{1}{p(z)} \sum_{y \in \mathcal{Y}} p^{(m-1)}(z|y)p(x,y)$

4: $\quad d(y,z) \leftarrow D\big(p(x|y) \| p(x|z)\big)$

5: $\quad p^{(m)}(z|y) \leftarrow \frac{p(z)\exp\big(-\beta d(y,z)\big)}{\sum_{z' \in \mathcal{Z}} p(z')\exp\big(-\beta d(y,z')\big)}$

6: $\quad \overline{d}^{(m)} \leftarrow \sum_{y \in \mathcal{Y}} p(y) \sum_{z \in \mathcal{Z}} p^{(m)}(z|y)d(y,z)$

7: $\quad \eta \leftarrow \big(\overline{d}^{(m-1)} - \overline{d}^{(m)}\big)/\overline{d}^{(m)}$

8: $\quad m \leftarrow m + 1$

9: **end while**

10: $p(z|y) \leftarrow p^{(m-1)}(z|y)$

Output: optimized probabilistic mapping $p(z|y)$

be carried out by alternatingly iterating (2.129)-(2.131) [97, Theorem 5]. These alternating iterations are known as the iterative IB algorithm which is summarized in Algorithm 2.2. The iterative IB algorithm converges to a locally optimal solution of (2.138). Hence, the resulting mapping $p(z|y)$ depends on the initialization $p^{(0)}(z|y)$. To avoid bad local optima, it may be necessary to repeatedly run Algorithm 2.2 and retain the best solution. Algorithm 2.2 terminates after M iterations or if the relative decrease of the average distortion $\overline{d}$ is below ε. Finally, we note that a coding theorem for the IB problem in (2.128) is given in [31].

3

Blind Performance Estimation for Bayesian Detectors

In this chapter, we study soft-information-based blind performance estimation for Bayesian detectors. The problem setting and relevant background are discussed in Section 3.1. As a motivating example, we present blind bit error probability estimation for Gaussian channels in Section 3.2. In Section 3.3, we study the properties of log-likelihood ratios (LLRs) which we shall use to derive blind estimators in the binary case. However, our approach is neither limited to the case of binary hypotheses, nor does it make any Gaussian assumptions. In Section 3.4, we consider blind estimators for several performance criteria of Bayesian detectors. The mean-square error (MSE) performance of the proposed estimators is evaluated in Section 3.5. In Section 3.6, we derive the Cramér-Rao lower bound (CRLB) for bit error probability estimation in the Gaussian case and we prove that an efficient estimator does not exist in this setting. Application examples for the proposed estimators are discussed in Section 3.7. We conclude this chapter with a discussion of our results in Section 3.8.

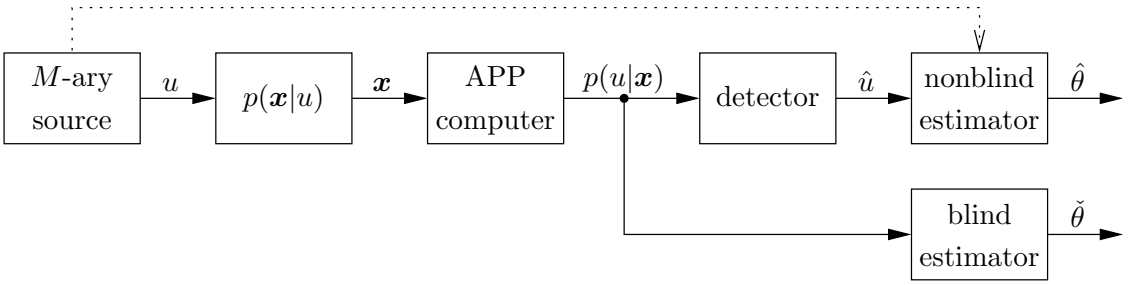


Figure 3.1: M-ary Bayesian hypothesis test with nonblind and blind estimation of some parameter θ.

3.1 Introduction and Background

Figure 3.1 depicts the setup we consider in this chapter. Let $u \in \mathcal{U}$ denote the output of an M-ary source, i.e., $|\mathcal{U}| = M$, with known prior probabilities $\mathbb{P}\{\mathsf{u} = u\}$. The observation $\boldsymbol{x}$ depends on u via the probabilistic mapping $p(\boldsymbol{x}|u)$ which is known for each u. We identify each u with one hypothesis, i.e., we consider an M-ary hypothesis test (cf. Section 2.4). Throughout, we assume that x is a continuous random variable with probability density function (pdf) $p(\boldsymbol{x})$. However, our results also hold in case x is a discrete random variable. In this case, pdfs are replaced by probability mass functions and integrals are replaced by sums. An optimal Bayesian detector (cf. (2.66)) computes its decision $\hat{u} \in \mathcal{U}$ based on the *a posteriori* probabilities (APPs) $\mathbb{P}\{\mathsf{u} = u|\mathsf{x} = \boldsymbol{x}\}$. In this context, we denote by θ a parameter that we shall estimate. An example for θ is the error probability of the detector.

We consider two methods for the estimation of θ. The first method uses the output of the detector together with the true source output u to produce the estimate $\hat{\theta}$. We call this approach *nonblind estimation* since it requires the source output u. Clearly, nonblind estimation can only be performed in computer simulations or when u is available as training data. The second method is referred to as *blind estimation* and uses the APPs $\mathbb{P}\{\mathsf{u} = u|\mathsf{x} = \boldsymbol{x}\}$, $u \in \mathcal{U}$, to produce the estimate $\check{\theta}$. A major advantage of blind estimation is that it does not require the source output u (which justifies its name). We note that performance estimation is relevant because an analytical performance evaluation is often infeasible and performance bounds may not always provide sufficient insight. Additionally, blind estimation is not restricted to Monte Carlo simulations and can thus be used for online performance evaluation of soft-information-based detectors.

In this context, it is natural to compare the performance of nonblind and blind estimation. For the error probability of binary convolutional codes, such a comparison is given in [67]. Blind bit error probability estimation in terms of LLRs is considered in [43, 57]. APP-based quality estimation for source-channel coded transmissions is studied in [96]. Blind estimation of mutual information with applications to extrinsic information transfer charts is considered in [58]. We note that the work presented in this chapter goes substantially beyond the previous work mentioned above.

Before we begin the discussion of blind estimators, we next present an example which addresses bit error probability estimation in the Gaussian case.

3.2 A Motivating Example

In this example we consider the model[1]

$$\mathbf{x} = \mathbf{1}_N \mathsf{u} + \mathbf{w}, \tag{3.1}$$

where $\mathbf{1}_N$ is the length-N all-ones vector, and $\mathsf{u} \in \{-1, 1\}$ is equally likely and independent of the noise $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$. In this case, the maximum *a posteriori* (MAP) detector is

$$\hat{u}(\boldsymbol{x}) = \begin{cases} 1, & L_{\mathsf{u}}(\boldsymbol{x}) > 0 \\ -1, & L_{\mathsf{u}}(\boldsymbol{x}) \leq 0 \end{cases}, \tag{3.2}$$

where

$$L_{\mathsf{u}}(\boldsymbol{x}) = \sum_{n=1}^{N} L_{\mathsf{u}}(x_n) = \sum_{n=1}^{N} \log \frac{\mathbb{P}\{\mathsf{u}=1|\mathsf{x}_n=x_n\}}{\mathbb{P}\{\mathsf{u}=-1|\mathsf{x}_n=x_n\}} = \frac{2}{\sigma^2} \sum_{n=1}^{N} x_n = \frac{2}{\sigma^2/N} \overline{x} \tag{3.3}$$

is the the posterior LLR for u. Here, we use the shorthand notation $\overline{x} \triangleq \frac{1}{N} \sum_{n=1}^{N} x_n$ for the arithmetic mean of $x_1, \ldots, x_N$. Due to (3.3), the LLR is conditionally Gaussian with

$$\mu_{\mathsf{L_u}|u} = \mathbb{E}\{\mathsf{L_u}|\mathsf{u}=u\} = \frac{2}{\sigma^2/N} u, \quad \text{and} \quad \sigma^2_{\mathsf{L_u}|u} = \mathrm{var}\{\mathsf{L_u}|\mathsf{u}=u\} = 2|\mu_{\mathsf{L_u}|u}|, \quad u \in \{-1, 1\}. \tag{3.4}$$

We note that $\sigma^2_{\mathsf{L_u}|u}$ does not depend on u and thus we have $\sigma^2_{\mathsf{L_u}} = \sigma^2_{\mathsf{L_u}|u}$. For the sake of notational simplicity, we use $\sigma^2_{\mathsf{L_u}}$ instead of $\sigma^2_{\mathsf{L_u}|u}$ in what follows.

Next, we compute the error probability conditioned on an observation $\boldsymbol{x}$ which we denote by $P_e(\boldsymbol{x})$. We have

$$P_e(\boldsymbol{x}) \triangleq \mathbb{P}\{\mathsf{u} \neq \hat{u}(\mathsf{x}) | \mathsf{x} = \boldsymbol{x}\}. \tag{3.5}$$

Rewriting (3.5) using Bayes' rule yields

$$P_e(\boldsymbol{x}) = \frac{p\big(\boldsymbol{x}|\mathsf{u} \neq \hat{u}(\boldsymbol{x})\big)\mathbb{P}\{\mathsf{u} \neq \hat{u}(\boldsymbol{x})\}}{p(\boldsymbol{x})} \tag{3.6}$$

$$= \frac{\frac{1}{2} p\big(\boldsymbol{x}|\mathsf{u} = -\hat{u}(\boldsymbol{x})\big)}{\frac{1}{2} p(\boldsymbol{x}|\mathsf{u}=1) + \frac{1}{2} p(\boldsymbol{x}|\mathsf{u}=-1)} \tag{3.7}$$

$$= \frac{1}{1 + \exp\big(|L_{\mathsf{u}}(\boldsymbol{x})|\big)}. \tag{3.8}$$

In (3.7), we have used the fact that $\mathsf{u} \neq \hat{u}(\boldsymbol{x})$ implies $\mathsf{u} = -\hat{u}(\boldsymbol{x})$ if $L_{\mathsf{u}}(\boldsymbol{x}) \neq 0$, and $\mathbb{P}\{\mathsf{u} \neq \hat{u}(\boldsymbol{x})\} = 1/2$ since $\mathsf{u} \in \{-1, 1\}$ is equally likely. The relation between the conditional error probability $P_e(\boldsymbol{x})$ and the LLR $L_{\mathsf{u}}(\boldsymbol{x})$ in (3.8) justifies and quantifies the reliability

[1] We note that the scalar model $\widetilde{\mathsf{x}} = \mathsf{u} + \widetilde{\mathsf{w}}$, with $\widetilde{\mathsf{w}} \sim \mathcal{N}(0, \sigma^2/N)$ independent of u, is equivalent to (3.1).

interpretation of LLRs. We note that $|L_{\mathsf{u}}(\boldsymbol{x})| = 0$ implies $P_e(\boldsymbol{x}) = 1/2$ and $|L_{\mathsf{u}}(\boldsymbol{x})| = \infty$ yields $P_e(\boldsymbol{x}) = 0$.

Using (3.4), (3.5), and (3.8), we write the unconditional error probability P_e as follows:

$$P_e = \mathbb{E}\left\{\frac{1}{1+\exp(|L_{\mathsf{u}}(\mathbf{x})|)}\right\} \tag{3.9}$$

$$= \int_{-\infty}^{\infty} \frac{p(L_{\mathsf{u}})}{1+\exp(|L_{\mathsf{u}}|)} \mathrm{d}L_{\mathsf{u}} \tag{3.10}$$

$$= \frac{1}{2}\int_{-\infty}^{\infty} \frac{p(L_{\mathsf{u}}|\mathsf{u}=1) + p(L_{\mathsf{u}}|\mathsf{u}=-1)}{1+\exp(|L_{\mathsf{u}}|)} \mathrm{d}L_{\mathsf{u}} \tag{3.11}$$

$$= \frac{1}{2\sqrt{2\pi\sigma^2_{\mathsf{L_u}}}} \int_{-\infty}^{0} \frac{\exp\left(-\frac{1}{2\sigma^2_{\mathsf{L_u}}}(L_{\mathsf{u}} - \sigma^2_{\mathsf{L_u}}/2)^2\right) + \exp\left(-\frac{1}{2\sigma^2_{\mathsf{L_u}}}(L_{\mathsf{u}} + \sigma^2_{\mathsf{L_u}}/2)^2\right)}{1+\exp(-L_{\mathsf{u}})} \mathrm{d}L_{\mathsf{u}}$$

$$+ \frac{1}{2\sqrt{2\pi\sigma^2_{\mathsf{L_u}}}} \int_{0}^{\infty} \frac{\exp\left(-\frac{1}{2\sigma^2_{\mathsf{L_u}}}(L_{\mathsf{u}} - \sigma^2_{\mathsf{L_u}}/2)^2\right) + \exp\left(-\frac{1}{2\sigma^2_{\mathsf{L_u}}}(L_{\mathsf{u}} + \sigma^2_{\mathsf{L_u}}/2)^2\right)}{1+\exp(L_{\mathsf{u}})} \mathrm{d}L_{\mathsf{u}} \tag{3.12}$$

$$= \frac{1}{2\sqrt{2\pi\sigma^2_{\mathsf{L_u}}}} \left[\int_{-\infty}^{0} \exp\left(-\frac{(L_{\mathsf{u}} - \sigma^2_{\mathsf{L_u}}/2)^2}{2\sigma^2_{\mathsf{L_u}}}\right) \mathrm{d}L_{\mathsf{u}} + \int_{0}^{\infty} \exp\left(-\frac{(L_{\mathsf{u}} + \sigma^2_{\mathsf{L_u}}/2)^2}{2\sigma^2_{\mathsf{L_u}}}\right) \mathrm{d}L_{\mathsf{u}}\right] \tag{3.13}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2_{\mathsf{L_u}}}} \int_{0}^{\infty} \exp\left(-\frac{1}{2\sigma^2_{\mathsf{L_u}}}(L_{\mathsf{u}} + \sigma^2_{\mathsf{L_u}}/2)^2\right) \mathrm{d}L_{\mathsf{u}}. \tag{3.14}$$

In (3.13), we have used the following result for Gaussian random variables.

Proposition 3.1. *Let* $\mathsf{x}_1 \sim \mathcal{N}(\mu, \sigma^2)$ *and* $\mathsf{x}_2 \sim \mathcal{N}(-\mu, \sigma^2)$ *be Gaussian random variables with pdfs* $p_{\mathsf{x}_1}(x_1)$ *and* $p_{\mathsf{x}_2}(x_2)$*, respectively. If and only if* $|\mu| = \sigma^2/2$*, then*

$$\frac{p_{\mathsf{x}_1}(z) + p_{\mathsf{x}_2}(z)}{1+e^{\pm z}} \tag{3.15}$$

is the pdf of a Gaussian random variable with mean $\mp\sigma^2/2$ *and variance* σ^2*.*

Proof: Proposition 3.1 can be shown by evaluating (3.15) and comparing it to the pdf of a Gaussian random variable. ■

Using the Q-function

$$Q(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} \exp(-t^2/2)\mathrm{d}t, \tag{3.16}$$

we can rewrite the tail probability in (3.14) as follows:

$$P_e = Q\left(\frac{\sigma_{\mathsf{L_u}|\mathsf{u}}}{2}\right) = Q\left(\frac{1}{\sqrt{\sigma^2/N}}\right) = \frac{1}{2}\operatorname{erfc}\left(\frac{1}{\sqrt{2\sigma^2/N}}\right). \tag{3.17}$$

We recognize (3.17) as the error probability of uncoded binary phase-shift keying (BPSK) transmission over an additive white Gaussian noise (AWGN) channel with signal-to-noise ratio (SNR) $(\sigma^2/N)^{-1/2}$ (see, e.g., [34, Section 6.1]).

Given K independent and identically distributed (iid) observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$, an unbiased and consistent blind estimator for the unconditional error probability P_e is (cf. (3.9))

$$\check{P}_e = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{1 + \exp(|L_{\mathsf{u}}(\boldsymbol{x}_k)|)}. \tag{3.18}$$

The corresponding nonblind estimator is given by

$$\hat{P}_e = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\{u_k \neq \hat{u}(\boldsymbol{x}_k)\}, \tag{3.19}$$

where u_k, $k = 1, \ldots, K$, are the true source outputs corresponding to the observations $\boldsymbol{x}_k$. We note that (3.18) is the arithmetic mean of K values in $[0, 1/2]$ and the knowledge of u_k, $k = 1, \ldots, K$, cannot improve the blind estimator $\check{P}_e$. In contrast, the nonblind estimator $\hat{P}_e$ averages K numbers which are either 0 or 1 (cf. (3.19)) and can therefore be expected to perform worse than $\check{P}_e$ (see Section 3.5). Consequently, we have $\check{P}_e \in [0, 1/2]$ and $\hat{P}_e \in [0, 1]$ (note that $P_e \in [0, 1/2]$, cf. (3.92)).

In the model (3.1), the error probability P_e is determined by σ^2/N (cf. (3.17)). Therefore, a blind estimator for P_e could first compute an estimate of σ^2 and then use (3.17) to obtain an estimate for P_e. Specifically, we may want to use the estimator

$$\check{\check{P}}_e = Q\Big(\big(\check{\check{\sigma}}^2/N\big)^{-1/2}\Big), \tag{3.20}$$

where

$$\check{\check{\sigma}}^2/N = \left[\frac{1}{K} \sum_{k=1}^{K} \overline{x}_k^2 - 1\right]^+. \tag{3.21}$$

However, $\check{\check{P}}_e$ is a biased estimator and its MSE performance is poor for small values of P_e.

Before we discuss blind estimators in more detail in Section 3.4, we next study the properties of LLRs. These properties are essential in the derivation of blind estimators in the binary case ($M = 2$).

3.3 Properties of Log-Likelihood Ratios

In this section, we consider the binary case with $\mathcal{U} = \{-1, 1\}$. As discussed in Section 2.4, the optimal Bayesian test in the binary case is a likelihood ratio test. Therefore, (log-)likelihood ratios play a central role in the binary case. We next study the properties of LLRs which we shall subsequently use to derive blind performance estimators.

We first recall the definition of the posterior LLR for a binary random variable $\mathsf{u} \in \{-1, 1\}$ from Subsection 2.6.1. We have

$$L_{\mathsf{u}}(\boldsymbol{x}) = \log \frac{\mathbb{P}\{\mathsf{u}=1|\mathsf{x}=\boldsymbol{x}\}}{\mathbb{P}\{\mathsf{u}=-1|\mathsf{x}=\boldsymbol{x}\}} = \log \frac{p(\boldsymbol{x}|\mathsf{u}=1)}{p(\boldsymbol{x}|\mathsf{u}=-1)} + \log \frac{\mathbb{P}\{\mathsf{u}=1\}}{\mathbb{P}\{\mathsf{u}=-1\}} = L_{\mathsf{u}}^{c}(\boldsymbol{x}) + L_{\mathsf{u}}^{a}. \tag{3.22}$$

Here, $L_{\mathsf{u}}^{c}(\boldsymbol{x})$ is the LLR due to the observation $\boldsymbol{x}$ and L_{u}^{a} is the prior LLR. The posterior probability of u can be expressed in terms of $L_{\mathsf{u}}(\boldsymbol{x})$ as follows:

$$\mathbb{P}\{\mathsf{u}=u|\mathsf{x}=\boldsymbol{x}\} = \frac{1}{1+e^{-uL_{\mathsf{u}}(\boldsymbol{x})}}, \quad u \in \{-1, 1\}. \tag{3.23}$$

A basic property of LLRs is that $L_{\mathsf{u}}\big(L_{\mathsf{u}}(\boldsymbol{x})\big) = L_{\mathsf{u}}(\boldsymbol{x})$. We summarize this result in the following Lemma.

Lemma 3.2. *The posterior LLR for* u *given the observation* $L_{\mathsf{u}}(\boldsymbol{x})$ *equals* $L_{\mathsf{u}}(\boldsymbol{x})$*. Specifically, we have*

$$L_{\mathsf{u}}\big(L_{\mathsf{u}}(\boldsymbol{x})\big) = \log \frac{p(L_{\mathsf{u}}(\boldsymbol{x})|\mathsf{u}=1)}{p(L_{\mathsf{u}}(\boldsymbol{x})|\mathsf{u}=-1)} + L_{\mathsf{u}}^{a} = L_{\mathsf{u}}(\boldsymbol{x}). \tag{3.24}$$

Proof: See Appendix A.1. ∎

Rewriting (3.24) yields the following relation between the conditional distributions of the posterior LLR $L_{\mathsf{u}}(\boldsymbol{x})$:

$$p(L_{\mathsf{u}}|\mathsf{u}=1) = \exp(L_{\mathsf{u}} - L_{\mathsf{u}}^{a}) p(L_{\mathsf{u}}|\mathsf{u}=-1). \tag{3.25}$$

This relationship is sometimes referred to as the "consistency condition" (see, e.g., [38]). LLRs are always consistent in the sense that (3.25) is fulfilled. However, approximate LLRs, e.g., those computed by an iterative decoder (cf. Subsection 2.6.5), may not be consistent. The case of approximate LLRs is discussed in Section 3.7 in more detail.

As a consequence of Lemma 3.2, conditioning on the LLR instead of the observation in (3.23) does not change the result. Therefore, we can write the posterior probability of u as

$$\mathbb{P}\{\mathsf{u}=u|\mathsf{L}_{\mathsf{u}}=L_{\mathsf{u}}\} = \frac{1}{1+e^{-uL_{\mathsf{u}}}}, \quad u \in \{-1, 1\}. \tag{3.26}$$

Using Bayes' rule and (3.26) we write the conditional distribution of L_{u} as follows:

$$p(L_{\mathsf{u}}|\mathsf{u}=u) = \mathbb{P}\{\mathsf{u}=u|\mathsf{L}_{\mathsf{u}}=L_{\mathsf{u}}\} \frac{p(L_{\mathsf{u}})}{\mathbb{P}\{\mathsf{u}=u\}} \tag{3.27}$$

$$= \frac{1}{1+e^{-uL_{\mathsf{u}}}} \frac{p(L_{\mathsf{u}})}{\mathbb{P}\{\mathsf{u}=u\}}, \quad u \in \{-1, 1\}. \tag{3.28}$$

We note that (3.28) allows us to express the conditional LLR distributions $p(L_{\mathsf{u}}|\mathsf{u}=u)$ in terms of the unconditional LLR distribution $p(L_{\mathsf{u}})$. More generally, the three distributions $p(L_{\mathsf{u}})$, $p(L_{\mathsf{u}}|\mathsf{u}=1)$, and $p(L_{\mathsf{u}}|\mathsf{u}=-1)$ are related through (3.28) such that any one of them is sufficient to express the other two. The following result is a consequence of (3.28).

Proposition 3.3. *The conditional and unconditional expectations of functions of the posterior LLR* L_u *are related as follows:*

$$\mathbb{E}\{g(\mathsf{L}_\mathsf{u})|\mathsf{u}=u\} = \frac{1}{\mathbb{P}\{\mathsf{u}=u\}}\mathbb{E}\left\{\frac{g(\mathsf{L}_\mathsf{u})}{1+e^{-u\mathsf{L}_\mathsf{u}}}\right\}, \quad u \in \{-1,1\} \tag{3.29}$$

$$\mathbb{E}\{g(\mathsf{L}_\mathsf{u})\} = \mathbb{P}\{\mathsf{u}=u\}\mathbb{E}\left\{g(\mathsf{L}_\mathsf{u})(1+e^{-u\mathsf{L}_\mathsf{u}})|\mathsf{u}=u\right\}, \quad u \in \{-1,1\}. \tag{3.30}$$

Furthermore, we have

$$\mathbb{E}\{\mathsf{u}g(\mathsf{L}_\mathsf{u})\} = \mathbb{E}\left\{\frac{g(\mathsf{L}_\mathsf{u})}{1+e^{-\mathsf{L}_\mathsf{u}}}\right\} - \mathbb{E}\left\{\frac{g(\mathsf{L}_\mathsf{u})}{1+e^{\mathsf{L}_\mathsf{u}}}\right\} = \mathbb{E}\{g(\mathsf{L}_\mathsf{u})\tanh(\mathsf{L}_\mathsf{u}/2)\}. \tag{3.31}$$

Proof: The relations in (3.29) and (3.30) follow by multiplying (3.28) with $g(L_\mathsf{u})$ and taking the expectation. Equation (3.31) follows by applying (3.29) to the right-hand side of the following equation:

$$\mathbb{E}\{\mathsf{u}g(\mathsf{L}_\mathsf{u})\} = \mathbb{E}\{g(\mathsf{L}_\mathsf{u})|\mathsf{u}=1\}\mathbb{P}\{\mathsf{u}=1\} - \mathbb{E}\{g(\mathsf{L}_\mathsf{u})|\mathsf{u}=-1\}\mathbb{P}\{\mathsf{u}=-1\}. \tag{3.32}$$

■

With $g(\mathsf{L}_\mathsf{u}) = \mathsf{L}_\mathsf{u}^k$, (3.29) and (3.30) provide relations between the conditional and the unconditional moments of L_u. The next result considers the special case $g(\mathsf{L}_\mathsf{u}) \equiv 1$.

Proposition 3.4. *The prior probabilities can be expressed as follows:*

$$\mathbb{P}\{\mathsf{u}=u\} = \mathbb{E}\left\{\frac{1}{1+e^{-u\mathsf{L}_\mathsf{u}}}\right\} = \left(\mathbb{E}\{1+e^{-u\mathsf{L}_\mathsf{u}}|\mathsf{u}=u\}\right)^{-1}, \quad u \in \{-1,1\}. \tag{3.33}$$

Hence, we have $\mathbb{P}\{\mathsf{u}=1\} = \mathbb{P}\{\mathsf{u}=-1\} = 1/2$ *if and only if* $p(-L_\mathsf{u}) = p(L_\mathsf{u})$.

Proof: If $\mathbb{P}\{\mathsf{u}=1\} = \mathbb{P}\{\mathsf{u}=-1\} = 1/2$, then

$$\int_{-\infty}^{\infty} \frac{p(L_\mathsf{u})}{1+e^{L_\mathsf{u}}} \mathrm{d}L_\mathsf{u} = \int_{-\infty}^{\infty} \frac{p(L_\mathsf{u})}{1+e^{-L_\mathsf{u}}} \mathrm{d}L_\mathsf{u} \tag{3.34}$$

which implies $p(-L_\mathsf{u}) = p(L_\mathsf{u})$. Conversely, if $p(-L_\mathsf{u}) = p(L_\mathsf{u})$, then (3.34) holds and thus $\mathbb{P}\{\mathsf{u}=1\} = \mathbb{P}\{\mathsf{u}=-1\} = 1/2$. ■

3.3.1 Uniform Prior Distribution

We next specialize the above results to an even LLR distribution, or, equivalently, to a uniform prior distribution on u. In this case, the consistency condition reads

$$p(L_\mathsf{u}|\mathsf{u}=1) = e^{L_\mathsf{u}} p(L_\mathsf{u}|\mathsf{u}=-1). \tag{3.35}$$

The conditional LLR distributions are given by

$$p(L_{\mathsf{u}}|\mathsf{u}=u) = \frac{2p(L_{\mathsf{u}})}{1+e^{-uL_{\mathsf{u}}}}, \quad u \in \{-1,1\}. \tag{3.36}$$

Due to (3.36), the conditional LLR distributions are furthermore related as follows:

$$p(-L_{\mathsf{u}}|\mathsf{u}=u) = e^{-uL_{\mathsf{u}}}p(L_{\mathsf{u}}|\mathsf{u}=u) = p(L_{\mathsf{u}}|\mathsf{u}=-u), \quad u \in \{-1,1\}. \tag{3.37}$$

As discussed above, (3.33) yields $\mathbb{P}\{\mathsf{u}=u\} = 1/2$, $u \in \{-1,1\}$. For an even function $g(\cdot)$, we have (cf. (3.29))

$$\mathbb{E}\{g(\mathsf{L}_{\mathsf{u}})|\mathsf{u}=u\} = \mathbb{E}\{g(\mathsf{L}_{\mathsf{u}})\}, \quad u \in \{-1,1\}. \tag{3.38}$$

An odd function $g(\cdot)$ yields $\mathbb{E}\{g(\mathsf{L}_{\mathsf{u}})\} = 0$ and

$$\mathbb{E}\{g(\mathsf{L}_{\mathsf{u}})|\mathsf{u}=u\} = u\mathbb{E}\{g(\mathsf{L}_{\mathsf{u}})\tanh(\mathsf{L}_{\mathsf{u}}/2)\} = u\mathbb{E}\{g(\mathsf{L}_{\mathsf{u}})\tanh(\mathsf{L}_{\mathsf{u}}/2)|\mathsf{u}=u\}, \quad u \in \{-1,1\}. \tag{3.39}$$

We note that the second equality in (3.39) is due to the fact that $g(\mathsf{L}_{\mathsf{u}})\tanh(\mathsf{L}_{\mathsf{u}}/2)$ is an even function if $g(\cdot)$ is an odd function. For a general function $g(\cdot)$ we have

$$\mathbb{E}\{g(\mathsf{L}_{\mathsf{u}})\} = \frac{1}{2}\mathbb{E}\{g(\mathsf{L}_{\mathsf{u}}) + g(-\mathsf{L}_{\mathsf{u}})\} + \frac{1}{2}\underbrace{\mathbb{E}\{g(\mathsf{L}_{\mathsf{u}}) - g(-\mathsf{L}_{\mathsf{u}})\}}_{=0} \tag{3.40}$$

$$= \frac{1}{2}\mathbb{E}\{g(\mathsf{L}_{\mathsf{u}}) + g(-\mathsf{L}_{\mathsf{u}})\} \tag{3.41}$$

$$= \frac{1}{2}\mathbb{E}\{g(\mathsf{L}_{\mathsf{u}}) + g(-\mathsf{L}_{\mathsf{u}})|\mathsf{u}=u\}, \quad u \in \{-1,1\}, \tag{3.42}$$

where (3.41) is due to the fact that $g(\mathsf{L}_{\mathsf{u}}) - g(-\mathsf{L}_{\mathsf{u}})$ is an odd function, and (3.42) follows from the fact that $g(\mathsf{L}_{\mathsf{u}}) + g(-\mathsf{L}_{\mathsf{u}})$ is an even function (cf. (3.38)). For the moments of the LLR, $p(-L_{\mathsf{u}}) = p(L_{\mathsf{u}})$ implies that $\mathbb{E}\{\mathsf{L}_{\mathsf{u}}\} = 0$ and $\mathbb{E}\{\mathsf{L}_{\mathsf{u}}^2\} = \mathbb{E}\{\mathsf{L}_{\mathsf{u}}^2|\mathsf{u}=u\}$, $u \in \{-1,1\}$.

3.3.2 Soft Bits

Let $g(\cdot)$ be an invertible function. The conditional distributions of the transformed random variable $\Psi_{\mathsf{u}} = g(\mathsf{L}_{\mathsf{u}})$ are given by

$$p(\Psi_{\mathsf{u}}|\mathsf{u}=u) = \frac{1}{1+\exp\big(-ug^{-1}(\Psi_{\mathsf{u}})\big)}\frac{p(\Psi_{\mathsf{u}})}{\mathbb{P}\{\mathsf{u}=u\}}, \quad u \in \{-1,1\}. \tag{3.43}$$

A particularly important function of L_{u} is the *soft bit* Λ_{u} which is defined as the minimum MSE estimate of u given the observation $\boldsymbol{x}$, i.e., we have[2]

$$\Lambda_{\mathsf{u}} \triangleq \mathbb{E}\{\mathsf{u}|\mathsf{x}=\boldsymbol{x}\} = \mathbb{P}\{\mathsf{u}=1|\mathsf{L}_{\mathsf{u}}=L_{\mathsf{u}}\} - \mathbb{P}\{\mathsf{u}=-1|\mathsf{L}_{\mathsf{u}}=L_{\mathsf{u}}\} = \tanh(L_{\mathsf{u}}/2). \tag{3.44}$$

[2]Note that the soft bit also appears in the boxplus operation (cf. (2.95) and (2.97)).

The above results can be translated to the soft bit domain by noting that $L_{\mathsf{u}} = g^{-1}(\Lambda_{\mathsf{u}}) = 2\,\mathrm{atanh}(\Lambda_{\mathsf{u}})$ and thus $\big(1+\exp(-uL_{\mathsf{u}})\big)^{-1} = (1+u\Lambda_{\mathsf{u}})/2$. Hence, we can write the conditional soft bit distribution as follows (cf. (3.28)):

$$p(\Lambda_{\mathsf{u}}|\mathsf{u}=u) = \frac{1+u\Lambda_{\mathsf{u}}}{2}\frac{p(\Lambda_{\mathsf{u}})}{\mathbb{P}\{\mathsf{u}=u\}}, \quad u \in \{-1,1\}. \tag{3.45}$$

For an even soft bit distribution $p(\Lambda_{\mathsf{u}})$, (3.45) simplifies to

$$p(\Lambda_{\mathsf{u}}|\mathsf{u}=u) = (1+u\Lambda_{\mathsf{u}})p(\Lambda_{\mathsf{u}}), \quad u \in \{-1,1\}, \tag{3.46}$$

since $p(-\Lambda_{\mathsf{u}}) = p(\Lambda_{\mathsf{u}})$ implies $\mathbb{P}\{\mathsf{u}=1\} = 1/2$. In this case, the conditional moments of Λ_{u} are related as follows:

$$\mathbb{E}\{\Lambda_{\mathsf{u}}|\mathsf{u}=u\} = u\mathbb{E}\{\Lambda_{\mathsf{u}}^2\}, \tag{3.47}$$

$$\mathbb{E}\{\Lambda_{\mathsf{u}}^2|\mathsf{u}=u\} = \mathbb{E}\{\Lambda_{\mathsf{u}}^2\}, \tag{3.48}$$

$$\mathrm{var}\{\Lambda_{\mathsf{u}}|\mathsf{u}=u\} = \mathbb{E}\{\Lambda_{\mathsf{u}}^2\}(1-\mathbb{E}\{\Lambda_{\mathsf{u}}^2\}). \tag{3.49}$$

In the next section, we discuss blind estimation in a more general setting than in our example in Section 3.2. Specifically, we allow for general M-ary Bayesian detectors with nonuniform prior probabilities, non-Gaussian $p(\boldsymbol{x}|u)$, and we consider other parameters than the error probability.

3.4 Blind Estimators

The optimal Bayesian detector for $\mathsf{u} \in \mathcal{U}$ is of the form

$$\hat{u}(\boldsymbol{x}) = \arg\min_{\tilde{u}} R(\tilde{u}), \tag{3.50}$$

where $R(\tilde{u})$ denotes the Bayesian risk associated to the detector $\tilde{u}$ (cf. Section 2.4). In the binary case with $\mathsf{u} \in \{-1,1\}$, the optimal detector (3.50) compares the posterior LLR $L_{\mathsf{u}}(\boldsymbol{x})$ to a threshold γ, i.e., we have[3]

$$\hat{u}(L_{\mathsf{u}}) = \begin{cases} 1, & L_{\mathsf{u}} > \gamma \\ -1, & L_{\mathsf{u}} \leq \gamma \end{cases}. \tag{3.51}$$

In what follows, we propose blind estimators which use K iid observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$ to estimate performance-related parameters of Bayesian detectors. The observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$ correspond to the source outputs $u_1, \ldots, u_K$. The estimators proposed in this section are unbiased and consistent. A performance analysis (in terms of MSE) of these blind estimators is given in Section 3.5.

[3] In contrast to Section 2.4, we formulate the optimal binary detector (3.51) in terms of the *posterior* LLR. Therefore, the threshold γ in (3.51) is different from the threshold in Section 2.4.

3.4.1 False Alarm Probability

The false alarm probability of a binary Bayesian detector is given by

$$\begin{aligned} P_F &\triangleq \mathbb{P}\{\hat{u}(\mathsf{L_u})=-1|\mathsf{u}=1\} && (3.52)\\ &= \mathbb{P}\{\mathsf{L_u}\leq\gamma|\mathsf{u}=1\} && (3.53)\\ &= \int_{-\infty}^{\gamma} p(L_\mathsf{u}|\mathsf{u}=1)\mathrm{d}L_\mathsf{u} && (3.54)\\ &= \frac{1}{\mathbb{P}\{\mathsf{u}=1\}}\int_{-\infty}^{\gamma}\frac{1}{1+e^{-L_\mathsf{u}}}p(L_\mathsf{u})\mathrm{d}L_\mathsf{u} && (3.55)\\ &= \frac{1}{\mathbb{P}\{\mathsf{u}=1\}}\int_{-\infty}^{\infty}\frac{s(\gamma-L_\mathsf{u})}{1+e^{-L_\mathsf{u}}}p(L_\mathsf{u})\mathrm{d}L_\mathsf{u} && (3.56)\\ &= \frac{1}{\mathbb{P}\{\mathsf{u}=1\}}\mathbb{E}\left\{\frac{s(\gamma-\mathsf{L_u})}{1+e^{-\mathsf{L_u}}}\right\}, && (3.57)\end{aligned}$$

where $s(\cdot)$ denotes the unit step function. From (3.57) it follows that the false alarm probability is upper bounded as follows:

$$P_F \leq \min\left\{1, \frac{1}{\mathbb{P}\{\mathsf{u}=1\}(1+e^{-\gamma})}\right\}. \tag{3.58}$$

We note that the bound in (3.58) is sharp in the sense that for any choice of $\mathbb{P}\{\mathsf{u}=1\}$ and γ, there exists at least one distribution of the data such that (3.58) is satisfied with equality. Due to (3.57), a blind estimator for P_F is given by

$$\check{P}_F = \frac{1}{\mathbb{P}\{\mathsf{u}=1\}}\frac{1}{K}\sum_{k=1}^{K}\frac{s\big(\gamma-L_\mathsf{u}(\boldsymbol{x}_k)\big)}{1+e^{-L_\mathsf{u}(\boldsymbol{x}_k)}}. \tag{3.59}$$

A corresponding nonblind estimator for P_F is given by

$$\hat{P}_F = \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\mathbb{1}\{\hat{u}\big(L_\mathsf{u}(\boldsymbol{x}_k)\big)=-1\}, \tag{3.60}$$

where we have defined $\mathcal{K} \triangleq \{k \,|\, u_k = 1\}$. An important difference between $\check{P}_F$ and $\hat{P}_F$ is that the nonblind estimator requires $|\mathcal{K}| \geq 1$, which is in contrast to the blind estimator. If the source outputs $u_1, \ldots, u_K$ are training data, then the best choice for estimating P_F is $u_1 = \ldots = u_K = 1$.

In a communications setting, the false alarm probability is the conditional bit error probability given $\mathsf{u} = 1$. For the special case of a MAP detector (i.e., $\gamma = 0$) and a uniform prior distribution, the false alarm probability equals

$$P_F = \int_{-\infty}^{0}\frac{2p(L_\mathsf{u})}{1+e^{-L_\mathsf{u}}}\mathrm{d}L_\mathsf{u} = \int_{0}^{\infty}\frac{2p(L_\mathsf{u})}{1+e^{L_\mathsf{u}}}\mathrm{d}L_\mathsf{u} = \int_{-\infty}^{\infty}\frac{p(L_\mathsf{u})}{1+e^{|L_\mathsf{u}|}}\mathrm{d}L_\mathsf{u} = \mathbb{E}\left\{\frac{1}{1+e^{|\mathsf{L_u}|}}\right\}. \tag{3.61}$$

In this case, we have $P_F = P_M = P_e$ (cf. (3.91)) and a blind estimator for P_F is given by

$$\check{P}_F = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{1+e^{|L_{\mathsf{u}}(\boldsymbol{x}_k)|}}. \tag{3.62}$$

3.4.2 Detection Probability

Similar to the false alarm probability, we rewrite the detection probability $P_D \triangleq \mathbb{P}\{\hat{u}(\mathsf{L_u}) = -1 | \mathsf{u} = -1\}$ as

$$P_D = \frac{1}{\mathbb{P}\{\mathsf{u}=-1\}}\mathbb{E}\left\{\frac{s(\gamma-\mathsf{L_u})}{1+e^{\mathsf{L_u}}}\right\}. \tag{3.63}$$

A sharp lower bound for the detection probability is given by (cf. the upper bound for P_M in (3.70))

$$P_D \geq \max\left\{0, 1-\frac{1}{\mathbb{P}\{\mathsf{u}=-1\}(1+e^{\gamma})}\right\}. \tag{3.64}$$

The relation in (3.63) yields the following blind estimator for P_D:

$$\check{P}_D = \frac{1}{\mathbb{P}\{\mathsf{u}=-1\}}\frac{1}{K}\sum_{k=1}^{K}\frac{s\big(\gamma - L_{\mathsf{u}}(\boldsymbol{x}_k)\big)}{1+e^{L_{\mathsf{u}}(\boldsymbol{x}_k)}}. \tag{3.65}$$

A nonblind estimator for P_D is given by

$$\hat{P}_D = \frac{1}{|\overline{\mathcal{K}}|}\sum_{k\in\overline{\mathcal{K}}}\mathbb{1}\{\hat{u}\big(L_{\mathsf{u}}(\boldsymbol{x}_k)\big) = -1\}, \tag{3.66}$$

where we have defined $\overline{\mathcal{K}} \triangleq \{k \,|\, u_k = -1\}$. We note that the estimators $\check{P}_F$ and $\check{P}_D$ enable binary Bayesian detectors to blindly estimate their receiver operating characteristic.

The special case of a MAP detector together with a uniform prior distribution yields $P_D = 1 - P_F = 1 - P_M = \mathbb{E}\{(1+e^{-|\mathsf{L_u}|})^{-1}\}$. Thus, a blind estimator for P_D is given by

$$\check{P}_D = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{1+e^{-|L_{\mathsf{u}}(\boldsymbol{x}_k)|}}. \tag{3.67}$$

We note that in this case $P_D = P_A = 1 - P_e$ (cf. (3.91)).

3.4.3 Acceptance Probability and Miss Probability

The acceptance and miss probabilities are related to the false alarm and detection probabilities by $P_A = 1 - P_F$ and $P_M = 1 - P_D$. Hence, we have

$$P_A = \frac{1}{\mathbb{P}\{\mathsf{u}=1\}}\mathbb{E}\left\{\frac{s(\mathsf{L_u}-\gamma)}{1+e^{-\mathsf{L_u}}}\right\}, \tag{3.68}$$

$$P_M = \frac{1}{\mathbb{P}\{\mathsf{u}=-1\}}\mathbb{E}\left\{\frac{s(\mathsf{L_u}-\gamma)}{1+e^{\mathsf{L_u}}}\right\}. \tag{3.69}$$

Using (3.69) we can upper bound P_M as follows:

$$P_M \leq \min\left\{1, \frac{1}{\mathbb{P}\{\mathsf{u}=-1\}(1+e^{\gamma})}\right\}. \tag{3.70}$$

The bound in (3.64) is obtained using (3.70). Similarly, (3.58) yields the following lower bound for P_A:

$$P_A \geq \max\left\{0, 1-\frac{1}{\mathbb{P}\{\mathsf{u}=1\}(1+e^{-\gamma})}\right\}. \tag{3.71}$$

The blind estimators for P_A and P_M can be derived directly from (3.59) and (3.65). In particular, we have

$$\check{P}_A = 1-\check{P}_F = \frac{1}{\mathbb{P}\{\mathsf{u}=1\}}\frac{1}{K}\sum_{k=1}^{K}\frac{s\big(L_{\mathsf{u}}(\boldsymbol{x}_k)-\gamma\big)}{1+e^{-L_{\mathsf{u}}(\boldsymbol{x}_k)}} \tag{3.72}$$

and

$$\check{P}_M = 1-\check{P}_D = \frac{1}{\mathbb{P}\{\mathsf{u}=-1\}}\frac{1}{K}\sum_{k=1}^{K}\frac{s\big(L_{\mathsf{u}}(\boldsymbol{x}_k)-\gamma\big)}{1+e^{L_{\mathsf{u}}(\boldsymbol{x}_k)}}. \tag{3.73}$$

Similarly, the corresponding nonblind estimators are $\hat{P}_A = 1-\hat{P}_F$ and $\hat{P}_M = 1-\hat{P}_D$.

In a communications setting, the miss probability is the conditional bit error probability given $\mathsf{u}=-1$. As mentioned above, for a MAP detector and a uniform prior distribution $\check{P}_A$ is given by (3.67) and $\check{P}_M$ is given by (3.62).

3.4.4 Conditional Error Probability

In the binary case, the conditional error probabilities are given by P_F and P_M. In the general M-ary case, we have

$$P_e(u) \triangleq \mathbb{P}\{\hat{u}(\mathsf{x})\neq\mathsf{u}|\mathsf{u}=u\} = \int_{\mathcal{X}} \mathbb{1}\{\hat{u}(\boldsymbol{x})\neq u\}p(\boldsymbol{x}|u)\mathrm{d}\boldsymbol{x} \tag{3.74}$$

$$= \frac{1}{P\{\mathsf{u}=u\}}\int_{\mathcal{X}} \mathbb{1}\{\hat{u}(\boldsymbol{x})\neq u\}\mathbb{P}\{\mathsf{u}=u|\mathsf{x}=\boldsymbol{x}\}p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{3.75}$$

$$= \frac{1}{P\{\mathsf{u}=u\}}\mathbb{E}\big\{\mathbb{1}\{\hat{u}(\mathsf{x})\neq u\}\mathbb{P}\{\mathsf{u}=u|\mathsf{x}\}\big\}, \quad u\in\mathcal{U}. \tag{3.76}$$

Due to (3.76), a blind estimator for $P_e(u)$ is given by

$$\check{P}_e(u) = \frac{1}{P\{\mathsf{u}=u\}}\frac{1}{K}\sum_{k=1}^{K}\mathbb{1}\{\hat{u}(\boldsymbol{x}_k)\neq u\}\mathbb{P}\{\mathsf{u}=u|\mathsf{x}=\boldsymbol{x}_k\}, \quad u\in\mathcal{U}. \tag{3.77}$$

The corresponding nonblind estimator for $P_e(u)$ equals

$$\hat{P}_e(u) = \frac{1}{|\mathcal{K}_u|}\sum_{k\in\mathcal{K}_u}\mathbb{1}\{\hat{u}(\boldsymbol{x}_k)\neq u\}, \quad u\in\mathcal{U}, \tag{3.78}$$

where we have defined $\mathcal{K}_u \triangleq \{k \,|\, u_k = u\}$. We note that (3.78) requires $|\mathcal{K}_u| \geq 1$ which is in contrast to the blind estimator in (3.77). For the special case of a MAP detector we replace $\hat{u}(\boldsymbol{x})$ by $\arg\max_{\tilde{u}\in\mathcal{U}} p(\tilde{u}|\boldsymbol{x})$ in (3.74)-(3.78).

3.4.5 Error Probability

The unconditional error probability $P_e = \mathbb{E}\{P_e(\mathsf{u})\}$ can be written as follows:

$$P_e = \sum_{u\in\mathcal{U}} \mathbb{E}\big\{\mathbb{1}\{\hat{u}(\mathsf{x}) \neq u\}\mathbb{P}\{\mathsf{u}=u|\mathsf{x}\}\big\} \tag{3.79}$$

$$= 1 - \sum_{u\in\mathcal{U}} \mathbb{E}\big\{\mathbb{1}\{\hat{u}(\mathsf{x}) = u\}\mathbb{P}\{\mathsf{u}=u|\mathsf{x}\}\big\} \tag{3.80}$$

$$= 1 - \mathbb{E}\big\{\mathbb{P}\{\mathsf{u}=\hat{u}(\mathsf{x})|\mathsf{x}\}\big\}. \tag{3.81}$$

For the special case of a MAP detector, we have

$$P_e = 1 - \mathbb{E}\big\{\max_{u\in\mathcal{U}} \mathbb{P}\{\mathsf{u}=u|\mathsf{x}\}\big\}. \tag{3.82}$$

Using (3.81), a blind estimators for P_e is given by

$$\check{P}_e = 1 - \frac{1}{K}\sum_{k=1}^{K} \mathbb{P}\{\mathsf{u}=\hat{u}(\mathsf{x})|\mathsf{x}=\boldsymbol{x}_k\}. \tag{3.83}$$

For a MAP detector, (3.83) can be rewritten as follows:

$$\check{P}_e = 1 - \frac{1}{K}\sum_{k=1}^{K} \max_{u\in\mathcal{U}} \mathbb{P}\{\mathsf{u}=u|\mathsf{x}=\boldsymbol{x}_k\}. \tag{3.84}$$

A corresponding nonblind estimator simply divides the number of error events by the number of samples, i.e., we have

$$\hat{P}_e = \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\{\hat{u}(\boldsymbol{x}_k) \neq u_k\}. \tag{3.85}$$

In the binary case, the error probability equals $P_e = P_F\mathbb{P}\{\mathsf{u}=1\} + P_M\mathbb{P}\{\mathsf{u}=-1\}$. We can thus rewrite P_e as follows:

$$P_e = \mathbb{E}\left\{\frac{s(\gamma - \mathsf{L}_\mathsf{u})}{1+e^{-\mathsf{L}_\mathsf{u}}} + \frac{s(\mathsf{L}_\mathsf{u} - \gamma)}{1+e^{\mathsf{L}_\mathsf{u}}}\right\}. \tag{3.86}$$

Using (3.30), P_e can be written as

$$P_e = \mathbb{E}\{s(\gamma - \mathsf{L}_\mathsf{u}) + s(\mathsf{L}_\mathsf{u} - \gamma)e^{-\mathsf{L}_\mathsf{u}}|\mathsf{u}=1\}\mathbb{P}\{\mathsf{u}=1\} \tag{3.87}$$

$$= \mathbb{E}\{s(\gamma - \mathsf{L}_\mathsf{u})e^{\mathsf{L}_\mathsf{u}} + s(\mathsf{L}_\mathsf{u} - \gamma)|\mathsf{u}=-1\}\mathbb{P}\{\mathsf{u}=-1\}. \tag{3.88}$$

From (3.87) and (3.88) we obtain the following sharp upper bound for P_e:

$$P_e \leq \min\big\{\mathbb{P}\{\mathsf{u}=1\}\max\{1,e^{-\gamma}\},\mathbb{P}\{\mathsf{u}=-1\}\max\{1,e^{\gamma}\}\big\}. \tag{3.89}$$

A weaker upper bound is given by

$$P_e \leq \frac{1}{1+e^{-|\gamma|}}, \tag{3.90}$$

where (3.89) and (3.90) are equivalent if and only if $\mathbb{P}\{\mathsf{u}=u\} = (1+e^{-u\gamma})^{-1}$. Specializing (3.86) and (3.89) to $\gamma = 0$, i.e., to a MAP detector, yields

$$P_e = \mathbb{E}\left\{\frac{1}{1+e^{|\mathsf{L}_\mathsf{u}|}}\right\} = \frac{1}{2}(1-\mathbb{E}\{|\Lambda_\mathsf{u}|\}), \tag{3.91}$$

and

$$P_e \leq \min\big\{\mathbb{P}\{\mathsf{u}=1\},\mathbb{P}\{\mathsf{u}=-1\}\big\} = \frac{1}{1+e^{|L_\mathsf{u}^a|}}. \tag{3.92}$$

The blind estimator (3.83) can be written in terms of LLRs as follows:

$$\check{P}_e = \frac{1}{K}\sum_{k=1}^{K}\left[\frac{s(\gamma - L_\mathsf{u}(\boldsymbol{x}_k))}{1+e^{-L_\mathsf{u}(\boldsymbol{x}_k)}} + \frac{s(L_\mathsf{u}(\boldsymbol{x}_k)-\gamma)}{1+e^{L_\mathsf{u}(\boldsymbol{x}_k)}}\right]. \tag{3.93}$$

For a MAP detector, (3.93) simplifies to

$$\check{P}_e = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{1+e^{|L_\mathsf{u}(\boldsymbol{x}_k)|}} = \frac{1}{2} - \frac{1}{2K}\sum_{k=1}^{K}|\Lambda_\mathsf{u}(\boldsymbol{x}_k)|. \tag{3.94}$$

We note that (3.94) equals the blind estimator for the bit error probability derived in Section 3.2. However, here we neither assumed that $p(\boldsymbol{x}|u) = \prod_n p(x_n|u)$ with conditionally Gaussian x_n, nor did we assume a uniform prior.

3.4.6 Block Error Probability

In communication scenarios, the block error probability is sometimes more relevant than the bit or symbol error probability. For a block of N source outputs $\mathsf{u}_1,\ldots,\mathsf{u}_N$ and corresponding data $\mathbf{x}_1,\ldots,\mathbf{x}_N$, we define the block error probability P_b as

$$P_b \triangleq \mathbb{P}\{\hat{u}(\mathbf{x}_1)\neq\mathsf{u}_1 \cup \cdots \cup \hat{u}(\mathbf{x}_N)\neq\mathsf{u}_N\}. \tag{3.95}$$

It is important to note that we consider separate detection of $\mathsf{u}_1,\ldots,\mathsf{u}_N$, although the source outputs may not be statistically independent. Hence, the block error probability is different from the error probability of (joint) detection of the "super-symbol" $\mathbf{u} = (\mathsf{u}_1 \ \ldots \ \mathsf{u}_N)^{\mathrm{T}}$. Moreover, the prior probabilities and the data model may be different for each source output. However, for the sake of notational simplicity we write $\hat{u}(\boldsymbol{x}_n)$ instead of $\hat{u}_n(\boldsymbol{x}_n)$ for the

detector of the nth source output. The block error probability (3.95) is bounded as follows:

$$P_b^- = \max_{n\in\{1,\ldots,N\}} \mathbb{P}\{\hat{u}(\mathsf{x}_n) \neq \mathsf{u}_n\} \leq P_b \leq \sum_{n=1}^{N} \mathbb{P}\{\hat{u}(\mathsf{x}_n) \neq \mathsf{u}_n\} = P_b^+. \tag{3.96}$$

We note that the bounds in (3.96) are sharp. The upper bound P_b^+ is satisfied with equality if all error events $\hat{u}(\mathsf{x}_n) \neq \mathsf{u}_n$, $n = 1, \ldots, N$, are mutually exclusive. Conversely, the lower bound P_b^- is satisfied with equality if all error events are equal (i.e., if all decisions are simultaneously (in)correct) or if at most one error event has positive probability. If all error events are statistically independent, the block error probability equals

$$P_b = 1 - \prod_{n=1}^{N} (1 - \mathbb{P}\{\hat{u}(\mathsf{x}_n) \neq \mathsf{u}_n\}) = 1 - \prod_{n=1}^{N} (1 - P_{e,n}). \tag{3.97}$$

Here, $P_{e,n}$ denotes the unconditional error probability for the nth source output.

A blind estimator for P_b of the form

$$\check{P}_b = 1 - \prod_{n=1}^{N} (1 - \check{P}_{e,n}) \tag{3.98}$$

converges to (3.97) and is thus unbiased if all error events are statistically independent. Here, $\check{P}_{e,n}$ corresponds to the blind estimator for the error probability in (3.83). If the error events are statistically dependent, then the estimator in (3.98) is biased. However, we always have

$$\check{P}_b^- = \max_{n\in\{1,\ldots,N\}} \check{P}_{e,n} \leq \check{P}_b \leq \sum_{n=1}^{N} \check{P}_{e,n} = \check{P}_b^+ \tag{3.99}$$

for all sample sizes $K \in \mathbb{N}$. We note that $\check{P}_b^+$ is an unbiased estimator for the upper bound P_b^+. Similarly, $\check{P}_b^-$ is an asymptotically unbiased estimator for the lower bound P_b^-. Hence, the value to which $\check{P}_b$ converges is always bounded as $P_b^- \leq \lim_{K\to\infty} \check{P}_b \leq P_b^+$. A nonblind estimator for P_b is given by

$$\hat{P}_b = 1 - \frac{1}{K} \sum_{k=1}^{K} \prod_{n=1}^{N} \mathbb{1}\{\hat{u}(\boldsymbol{x}_{n,k}) = u_n\}. \tag{3.100}$$

Here, $\boldsymbol{x}_{n,k}$ denotes the kth observation of the data corresponding to the nth source output. An alternative blind estimator for P_b is given by

$$\check{\hat{P}}_b = 1 - \frac{1}{K} \sum_{k=1}^{K} \prod_{n=1}^{N} \mathbb{P}\{\mathsf{u}_n = \hat{u}(\mathsf{x}_n) | \mathsf{x}_n = \boldsymbol{x}_{n,k}\}. \tag{3.101}$$

If all error events are statistically independent, the estimator in (3.101) is unbiased and

converges to (3.97). We note that $\check{\check{P}}_b$ is bounded as follows:

$$\check{\check{P}}_b^- = \frac{1}{K}\sum_{k=1}^{K} \max_{n\in\{1,\ldots,N\}} \mathbb{P}\{\mathsf{u}_n \neq \hat{u}(\mathbf{x}_n) | \mathbf{x}_n = \boldsymbol{x}_{n,k}\} \leq \check{\check{P}}_b \leq \sum_{n=1}^{N} \check{P}_{e,n} = \check{\check{P}}_b^+. \tag{3.102}$$

The usefulness of the bounds in (3.99) and (3.102) is confirmed by numerical results in Section 3.7.

3.4.7 Minimum MSE

Let $\hat{u}_{\mathrm{MMSE}}(\boldsymbol{x}) = \mathbb{E}\{\mathsf{u}|\mathbf{x}=\boldsymbol{x}\}$ denote the minimum MSE estimate of u given the observation $\boldsymbol{x}$. The corresponding minimum MSE equals

$$\begin{aligned}
\varepsilon_{\mathrm{MMSE}} &= \mathbb{E}\Big\{\big(\mathsf{u} - \hat{u}_{\mathrm{MMSE}}(\mathbf{x})\big)^2\Big\} && (3.103)\\
&= \mathbb{E}\{\mathsf{u}^2\} - 2\mathbb{E}\Big\{\mathsf{u}\sum\nolimits_{u\in\mathcal{U}} u\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\Big\} + \mathbb{E}\Big\{\Big(\sum\nolimits_{u\in\mathcal{U}} u\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\Big)^2\Big\} && (3.104)\\
&= \mathbb{E}\{\mathsf{u}^2\} - \sum_{u'\in\mathcal{U}}\sum_{u\in\mathcal{U}} u'u\mathbb{E}\big\{\mathbb{P}\{\mathsf{u}=u'|\mathbf{x}\}\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\big\}. && (3.105)
\end{aligned}$$

It is important to note that the minimum MSE does not depend on the detector $\hat{u}(\boldsymbol{x})$. However, $\varepsilon_{\mathrm{MMSE}}$ is a lower bound for the MSE $\mathbb{E}\big\{\big(\mathsf{u} - \hat{u}(\mathbf{x})\big)^2\big\}$, i.e., we have $\varepsilon_{\mathrm{MMSE}} \leq \mathbb{E}\{\big(\mathsf{u} - \hat{u}(\mathbf{x})\big)^2\}$. A blind estimator for $\varepsilon_{\mathrm{MMSE}}$ is given by

$$\check{\varepsilon}_{\mathrm{MMSE}} = \mathbb{E}\{\mathsf{u}^2\} - \frac{1}{K}\sum_{k=1}^{K}\sum_{u'\in\mathcal{U}}\sum_{u\in\mathcal{U}} u'u\mathbb{P}\{\mathsf{u}=u'|\mathbf{x}=\boldsymbol{x}_k\}\mathbb{P}\{\mathsf{u}=u|\mathbf{x}=\boldsymbol{x}_k\}. \tag{3.106}$$

In the binary case, we have $\hat{u}_{\mathrm{MMSE}}(\boldsymbol{x}) = \Lambda_{\mathsf{u}}(\boldsymbol{x}) = \tanh(L_{\mathsf{u}}(\boldsymbol{x})/2)$. Hence, the minimum MSE can be written as follows:

$$\begin{aligned}
\varepsilon_{\mathrm{MMSE}} &= \mathbb{E}\{\big(\mathsf{u} - \tanh(\mathsf{L}_{\mathsf{u}}/2)\big)^2\} && (3.107)\\
&= \mathbb{E}\{\mathsf{u}^2\} - 2\mathbb{E}\{\mathsf{u}\tanh(\mathsf{L}_{\mathsf{u}}/2)\} + \mathbb{E}\{\tanh^2(\mathsf{L}_{\mathsf{u}}/2)\} && (3.108)\\
&= 1 - \mathbb{E}\{\tanh^2(\mathsf{L}_{\mathsf{u}}/2)\} = 1 - \mathbb{E}\{\Lambda_{\mathsf{u}}^2\}. && (3.109)
\end{aligned}$$

In (3.109) we have used $\mathsf{u}^2 = 1$ and $\mathbb{E}\{\mathsf{u}\tanh(\mathsf{L}_{\mathsf{u}}/2)\} = \mathbb{E}\{\tanh^2(\mathsf{L}_{\mathsf{u}}/2)\}$ (cf. (3.31)). The blind estimator (3.106) simplifies as follows:

$$\check{\varepsilon}_{\mathrm{MMSE}} = 1 - \frac{1}{K}\sum_{k=1}^{K}\Lambda_{\mathsf{u}}^2(\boldsymbol{x}_k). \tag{3.110}$$

In the binary case, there is a connection between the MSE $\mathbb{E}\big\{\big(\mathsf{u} - \hat{u}(\mathbf{x})\big)^2\big\}$ and the error probability P_e. Indeed, we have

$$\mathbb{E}\big\{\big(\mathsf{u} - \hat{u}(\mathbf{x})\big)^2\big\} = 4P_F\mathbb{P}\{\mathsf{u}=1\} + 4P_M\mathbb{P}\{\mathsf{u}=-1\} = 4P_e. \tag{3.111}$$

Therefore, the minimum MSE lower bounds the error probability as follows:

$$P_e \geq \frac{1}{4}\varepsilon_{\mathrm{MMSE}} = \frac{1}{4}\left(1 - \mathbb{E}\{\Lambda_{\mathsf{u}}^2\}\right). \tag{3.112}$$

The MAP detector minimizes the left-hand side of (3.112), where $P_e = (1 - \mathbb{E}\{|\Lambda_{\mathsf{u}}|\})/2$. Together with (3.112), this yields

$$\mathbb{E}\{\Lambda_{\mathsf{u}}^2\} \geq 2\mathbb{E}\{|\Lambda_{\mathsf{u}}|\} - 1. \tag{3.113}$$

Finally, we note that estimating $\varepsilon_{\mathrm{MMSE}}$ requires the APPs $\mathbb{P}\{\mathsf{u}=u|\mathbf{x}=\boldsymbol{x}\}$. Hence, there is no nonblind estimator for the minimum MSE which operates on $\hat{u}(\boldsymbol{x})$.

3.4.8 Mutual Information and Conditional Entropy

Let us consider the mutual information $I(\mathsf{u};\mathbf{x})$ and the conditional entropy $H(\mathsf{u}|\mathbf{x})$. Since $I(\mathsf{u};\mathbf{x}) = H(\mathsf{u}) - H(\mathsf{u}|\mathbf{x})$, we focus on the conditional entropy in the following. We have

$$H(\mathsf{u}|\mathbf{x}) = -\sum_{u\in\mathcal{U}} \int_{\mathcal{X}} p(u,\boldsymbol{x}) \log_2 \mathbb{P}\{\mathsf{u}=u|\mathbf{x}=\boldsymbol{x}\} \mathrm{d}\boldsymbol{x} \tag{3.114}$$

$$= -\sum_{u\in\mathcal{U}} \mathbb{E}\big\{\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\} \log_2 \mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\big\}. \tag{3.115}$$

We note that $H(\mathsf{u}|\mathbf{x})$ and $I(\mathsf{u};\mathbf{x})$ do not depend on the detector $\hat{u}(\boldsymbol{x})$. A blind estimator for $H(\mathsf{u}|\mathbf{x})$ follows from (3.115) and is given by

$$\check{H}(\mathsf{u}|\mathbf{x}) = -\frac{1}{K}\sum_{k=1}^{K}\sum_{u\in\mathcal{U}} \mathbb{P}\{\mathsf{u}=u|\mathbf{x}=\boldsymbol{x}_k\} \log_2 \mathbb{P}\{\mathsf{u}=u|\mathbf{x}=\boldsymbol{x}_k\}. \tag{3.116}$$

Hence, $\check{I}(\mathsf{u};\mathbf{x}) = H(\mathsf{u}) - \check{H}(\mathsf{u}|\mathbf{x})$ is a blind estimator for the mutual information $I(\mathsf{u};\mathbf{x})$ (note that $H(\mathsf{u})$ is known since the source statistic $\mathbb{P}\{\mathsf{u}=u\}$, $u\in\mathcal{U}$, is known).

In the binary case, we rewrite (3.115) as follows:

$$H(\mathsf{u}|\mathbf{x}) = -\mathbb{E}\big\{\mathbb{P}\{\mathsf{u}=-1|\mathbf{x}\} \log_2 \mathbb{P}\{\mathsf{u}=-1|\mathbf{x}\} + \mathbb{P}\{\mathsf{u}=1|\mathbf{x}\} \log_2 \mathbb{P}\{\mathsf{u}=1|\mathbf{x}\}\big\} \tag{3.117}$$

$$= \mathbb{E}\left\{h_2\left(\frac{1}{1+e^{|\mathsf{L}_{\mathsf{u}}|}}\right)\right\} = \mathbb{E}\left\{h_2\left(\frac{1}{2}(1-|\Lambda_{\mathsf{u}}|)\right)\right\}, \tag{3.118}$$

where $h_2(\cdot)$ denotes the binary entropy function (cf. (2.17)). Since $h_2(\cdot)$ is a concave function, we can bound $H(\mathsf{u}|\mathbf{x})$ using the extended Jensen's inequality (cf. (2.14)) as follows:

$$2P_e^{\mathrm{MAP}} \leq H(\mathsf{u}|\mathbf{x}) \leq h_2(P_e^{\mathrm{MAP}}), \tag{3.119}$$

where P_e^{MAP} is the error probability of a MAP detector (cf. (3.91)). We note that the inequality $H(\mathsf{u}|\mathbf{x}) \leq h_2(P_e^{\mathrm{MAP}})$ can also be obtained using Fano's inequality [20, Section

2.10]. The bounds in (3.119) are sharp. Indeed, the lower bound is achieved by

$$p_{|\Lambda_{\mathsf{u}}|}(\Lambda_{\mathsf{u}}) = 2P_e^{\mathrm{MAP}}\delta(\Lambda_{\mathsf{u}}) + (1 - 2P_e^{\mathrm{MAP}})\delta(\Lambda_{\mathsf{u}} - 1). \tag{3.120}$$

Similarly, the upper bound is achieved by

$$p_{|\Lambda_{\mathsf{u}}|}(\Lambda_{\mathsf{u}}) = \delta(\Lambda_{\mathsf{u}} - 1 + 2P_e^{\mathrm{MAP}}). \tag{3.121}$$

We note that (3.120) corresponds to a binary erasure channel (BEC) and (3.121) corresponds to a binary symmetric channel (BSC). The bounds in (3.119) are reminiscent of bounds in the information combining literature [59, 60] where the BEC and the BSC are extreme cases, too.

An unbiased blind estimator for $H(\mathsf{u}|\mathsf{x})$ in the binary case is given by

$$\check{H}(\mathsf{u}|\mathsf{x}) = \frac{1}{K}\sum_{k=1}^{K} h_2\left(\frac{1}{1+e^{|L_{\mathsf{u}}(\boldsymbol{x}_k)|}}\right) = \frac{1}{K}\sum_{k=1}^{K} h_2\left(\frac{1}{2}(1 - |\Lambda_{\mathsf{u}}(\boldsymbol{x}_k)|)\right). \tag{3.122}$$

We note that the following inequalities hold for any sample size $K \in \mathbb{N}$:

$$2\check{P}_e^{\mathrm{MAP}} \leq \check{H}(\mathsf{u}|\mathsf{x}) \leq h_2(\check{P}_e^{\mathrm{MAP}}). \tag{3.123}$$

Therefore, $2\check{P}_e^{\mathrm{MAP}}$ is an unbiased estimator for the lower bound in (3.119) and $h_2(\check{P}_e^{\mathrm{MAP}})$ is an asymptotically unbiased estimator for the upper bound in (3.119). Due to (3.118), we can write the mutual information as

$$I(\mathsf{u};\mathsf{x}) = H(\mathsf{u}) - \mathbb{E}\left\{h_2\left(\frac{1}{1+e^{|\mathsf{L}_{\mathsf{u}}|}}\right)\right\} = H(\mathsf{u}) - \mathbb{E}\left\{h_2\left(\frac{1}{2}(1 - |\Lambda_{\mathsf{u}}|)\right)\right\}. \tag{3.124}$$

Hence, $\check{I}(\mathsf{u};\mathsf{x}) = H(\mathsf{u}) - \check{H}(\mathsf{u}|\mathsf{x})$ with the blind estimator for $H(\mathsf{u}|\mathsf{x})$ from (3.122). The mutual information and its blind estimate are bounded as follows:

$$H(\mathsf{u}) - h_2(P_e^{\mathrm{MAP}}) \leq I(\mathsf{u};\mathsf{x}) \leq H(\mathsf{u}) - 2P_e^{\mathrm{MAP}}, \tag{3.125}$$

$$H(\mathsf{u}) - h_2(\check{P}_e^{\mathrm{MAP}}) \leq \check{I}(\mathsf{u};\mathsf{x}) \leq H(\mathsf{u}) - 2\check{P}_e^{\mathrm{MAP}}. \tag{3.126}$$

The bounds in (3.125) are achieved by the distributions in (3.120) and (3.121).

Similarly, we can bound the error probability of the MAP detector by $I(\mathsf{u};\mathsf{x})$ and $H(\mathsf{u}|\mathsf{x})$ as follows:

$$h_2^{-1}\big(H(\mathsf{u}) - I(\mathsf{u};\mathsf{x})\big) = h_2^{-1}\big(H(\mathsf{u}|\mathsf{x})\big) \leq P_e^{\mathrm{MAP}} \leq \frac{1}{2}H(\mathsf{u}|\mathsf{x}) = \frac{1}{2}H(\mathsf{u}) - \frac{1}{2}I(\mathsf{u};\mathsf{x}), \tag{3.127}$$

where $h_2^{-1}\colon [0,1] \to [0,1/2]$ denotes the inverse of the binary entropy function. From (3.127) we can see that optimal processing in terms of the mutual information $I(\mathsf{u};\mathsf{x})$ minimizes the

upper and lower bounds for P_e^{MAP}. For an arbitrary detector $\hat{u}(\boldsymbol{x})$ we have (here, $\hat{\mathsf{u}} = \hat{u}(\mathbf{x})$)

$$H(\mathsf{u}|\hat{\mathsf{u}}) = -\sum_{\hat{u}\in\{-1,1\}} \mathbb{P}\{\hat{\mathsf{u}}=\hat{u}\} \sum_{u\in\{-1,1\}} \mathbb{P}\{\mathsf{u}=u|\hat{\mathsf{u}}=\hat{u}\}\log_2 \mathbb{P}\{\mathsf{u}=u|\hat{\mathsf{u}}=\hat{u}\} \tag{3.128}$$

$$= \sum_{\hat{u}\in\{-1,1\}} \mathbb{P}\{\hat{\mathsf{u}}=\hat{u}\}h_2(P_e) \tag{3.129}$$

$$= h_2(P_e). \tag{3.130}$$

Thus, $H(\mathsf{u}|\hat{\mathsf{u}})$ equals the upper bound for $H(\mathsf{u}|\mathbf{x})$ in (3.119) if and only if $P_e = P_e^{\mathrm{MAP}}$, i.e., if and only if $\hat{u}(\boldsymbol{x})$ is a MAP detector. The inequalities

$$H(\mathsf{u}|\mathbf{x}) \overset{(a)}{\leq} h_2(P_e^{\mathrm{MAP}}) \overset{(b)}{\leq} H(\mathsf{u}|\hat{\mathsf{u}}) = h_2(P_e) \tag{3.131}$$

are simultaneously satisfied with equality if (a) the data is distributed according to (3.121) and (b) a MAP detector is used.

3.5 Estimator Performance Analysis

The main goal of this section is to analyze the MSE performance of the blind estimators introduced in the previous section. Furthermore, where applicable we compare the blind estimators to the corresponding nonblind estimators and we show that in many cases the blind estimators are superior in terms of the MSE.

3.5.1 False Alarm Probability

The unbiasedness of the blind estimator

$$\check{P}_F = \frac{1}{K\mathbb{P}\{\mathsf{u}=1\}} \sum_{k=1}^{K} \frac{s\big(\gamma - L_{\mathsf{u}}(\boldsymbol{x}_k)\big)}{1+e^{-L_{\mathsf{u}}(\boldsymbol{x}_k)}} \tag{3.132}$$

follows directly from (3.57). The MSE of $\check{P}_F$ is given by

$$\mathrm{MSE}_{\check{P}_F}(P_F) = \mathbb{E}\{(\check{P}_F - P_F)^2\} = \frac{1}{K}\left(\frac{1}{\mathbb{P}\{\mathsf{u}=1\}}\right)^2 \mathbb{E}\left\{\left(\frac{s(\gamma-\mathsf{L}_{\mathsf{u}})}{1+e^{-\mathsf{L}_{\mathsf{u}}}}\right)^2\right\} - \frac{1}{K}P_F^2. \tag{3.133}$$

The expectation in (3.133) depends on the distribution of the data and in many cases (3.133) cannot be computed in closed form. However, we can bound the MSE by noting that

$$\left(\frac{s(\gamma-\mathsf{L}_{\mathsf{u}})}{1+e^{-\mathsf{L}_{\mathsf{u}}}}\right)^2 \leq \frac{1}{1+e^{-\gamma}} \frac{s(\gamma-\mathsf{L}_{\mathsf{u}})}{1+e^{-\mathsf{L}_{\mathsf{u}}}}. \tag{3.134}$$

More generally, for any $x \in [0, \alpha]$ we have $x^2 \leq \alpha x$. We use this inequality with $\alpha = (1 + e^{-\gamma})^{-1}$ to obtain (3.134). Using (3.134) in (3.133) yields

$$\mathrm{MSE}_{\check{P}_F}(P_F) \leq \frac{1}{K} \left(\frac{1}{\mathbb{P}\{\mathsf{u}=1\}} \right)^2 \frac{1}{1+e^{-\gamma}} \mathbb{E}\left\{ \frac{s(\gamma - \mathsf{L}_\mathsf{u})}{1+e^{-\mathsf{L}_\mathsf{u}}} \right\} - \frac{1}{K} P_F^2. \tag{3.135}$$

The expectation in (3.135) equals $\mathbb{P}\{\mathsf{u}=1\}P_F$ (cf. (3.57)) and we thus have

$$\mathrm{MSE}_{\check{P}_F}(P_F) \leq \frac{P_F}{K} \left(\frac{1}{\mathbb{P}\{\mathsf{u}=1\}(1+e^{-\gamma})} - P_F \right). \tag{3.136}$$

This upper bound for the MSE is sharp. Indeed, we have equality in (3.136) for any distribution of L_u that takes values only in $\{-\infty\} \cup [\gamma, \infty]$. However, for a specific distribution of the data, the upper bound in (3.136) may be rather loose. We note that the right-hand side of (3.136) is always nonnegative due to (3.58).

The nonblind estimator

$$\hat{P}_F = \frac{1}{\sum_{k=1}^K \mathbb{1}\{u_k=1\}} \sum_{k=1}^K \mathbb{1}\{\hat{u}\big(L_\mathsf{u}(\boldsymbol{x}_k)\big)=-1\}\mathbb{1}\{u_k=1\} \tag{3.137}$$

is unbiased and its MSE equals

$$\begin{aligned} \mathrm{MSE}_{\hat{P}_F}(P_F) &= \frac{1}{\kappa}\left(\mathbb{E}\big\{(\mathbb{1}\{\hat{u}\big(L_\mathsf{u}(\mathsf{x})\big)=-1\}\mathbb{1}\{u=1\})^2\big\} - P_F^2\right) &(3.138)\\ &= \frac{1}{\kappa}\left(\mathbb{E}\big\{\mathbb{1}\{\hat{u}\big(L_\mathsf{u}(\mathsf{x})\big)=-1\}\big\}\mathbb{1}\{u=1\} - P_F^2\right) = \frac{P_F}{\kappa}(1-P_F), &(3.139)\end{aligned}$$

where $\kappa = \sum_{k=1}^K \mathbb{1}\{u_k = 1\}$ denotes the number of source outputs which are equal to 1. Of course, the MSE in (3.139) is smallest if $\kappa = K$ which can be achieved by choosing $u_1 = \ldots = u_K = 1$ (if the source outputs are training data). The ratio of the MSEs in (3.139) and (3.133) is lower bounded as

$$\frac{\mathrm{MSE}_{\hat{P}_F}(P_F)}{\mathrm{MSE}_{\check{P}_F}(P_F)} \geq \frac{1-P_F}{\big(\mathbb{P}\{\mathsf{u}=1\}(1+e^{-\gamma})\big)^{-1} - P_F}. \tag{3.140}$$

The following result is a consequence of the MSE ratio in (3.140).

Proposition 3.5. *The blind estimator $\check{P}_F$ dominates the nonblind estimator $\hat{P}_F$ for any distribution of the data if and only if*

$$\mathbb{P}\{\mathsf{u}=1\}(1+e^{-\gamma}) \geq 1. \tag{3.141}$$

For specific distributions of the data, $\check{P}_F$ may dominate $\hat{P}_F$ even if (3.141) does not hold.

Proof: The lower bound (3.140) shows that $\mathrm{MSE}_{\check{P}_F}(P_F) \leq \mathrm{MSE}_{\hat{P}_F}(P_F)$ for all P_F if the inequality (3.141) is satisfied. ■

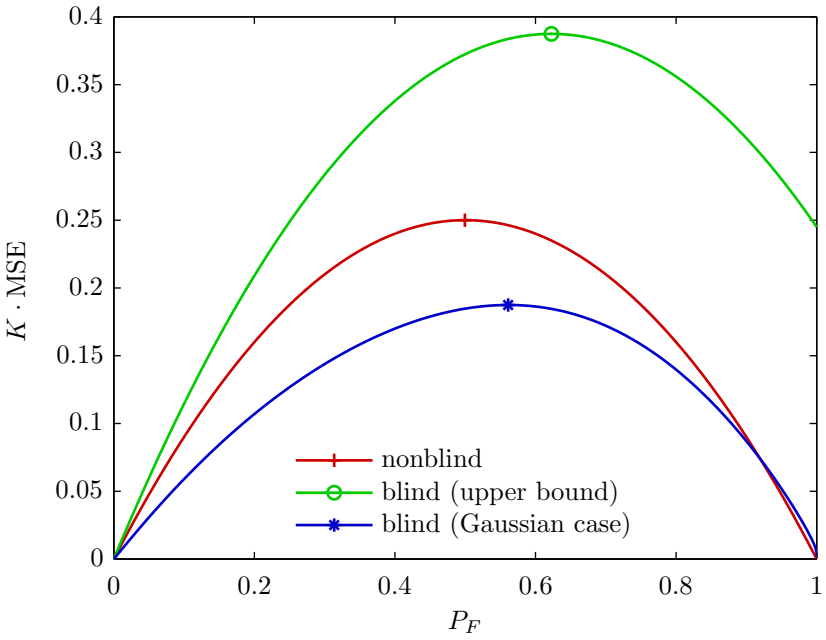


Figure 3.2: MSE versus P_F for $\mathbb{P}\{\mathsf{u}=1\} = 1/2$ and $\gamma = 1/2$. Comparison of $\mathrm{MSE}_{\hat{P}_F}(P_F)$ (cf. (3.139)), upper bound (3.136), and $\mathrm{MSE}_{\check{P}_F}(P_F)$ (cf. (3.133)) in the Gaussian case.

In the following example, we consider $\mathbb{P}\{\mathsf{u}=1\} = 1/2$ and $\gamma = 1/2$. The data is such that the LLR L_u is conditionally Gaussian, i.e., we have $\mathsf{L}_\mathsf{u}|u \sim \mathcal{N}(u\mu, 2\mu)$ with $\mu > 0$. In this case, the false alarm probability equals

$$P_F = 1 - Q\left(\frac{\gamma - \mu}{\sqrt{2\mu}}\right). \tag{3.142}$$

In Figure 3.2, we compare the MSE of the nonblind estimator (with $\kappa = K$) to the MSE of the blind estimator and the upper bound (3.136). Here, the upper bound is greater than (3.139) for all P_F since (3.141) does not hold. However, the MSE of the blind estimator is smaller than the MSE of the nonblind estimator for all $P_F \lesssim 0.92$, i.e., the blind estimator outperforms the nonblind estimator for all P_F of practical interest. The MSE ratio and the corresponding lower bound (3.140) are shown in Figure 3.4a.

Next, we let $\mathbb{P}\{\mathsf{u}=1\} = 0.7$. Then, $\mathsf{L}_\mathsf{u}|u \sim \mathcal{N}(u\mu + L_\mathsf{u}^a, 2\mu)$ with the prior LLR L_u^a and $\mu > 0$. In this case, the false alarm probability is given by

$$P_F = 1 - Q\left(\frac{\gamma - \mu - L_\mathsf{u}^a}{\sqrt{2\mu}}\right). \tag{3.143}$$

Due to (3.143), the maximum value of P_F is much smaller than 1 and some P_F may result from two different values of μ. Furthermore, the inequality (3.141) is now fulfilled and therefore the blind estimator dominates the nonblind estimator. A comparison of the MSEs is given in Figure 3.3. In the Gaussian case, almost every P_F corresponds to two different values of μ,

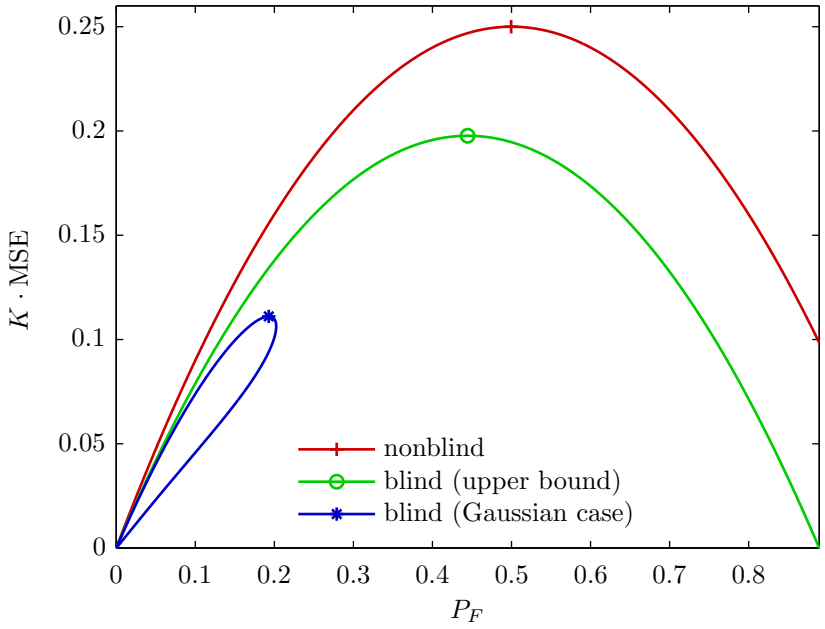


Figure 3.3: MSE versus P_F for $\mathbb{P}\{\mathsf{u}=1\} = 0.7$ and $\gamma = 1/2$. Comparison of $\mathrm{MSE}_{\hat{P}_F}(P_F)$ (cf. (3.139)), upper bound (3.136), and $\mathrm{MSE}_{\check{P}_F}(P_F)$ (cf. (3.133)) in the Gaussian case.

where the larger of the two values yields a smaller MSE. Hence, the blue curve ('$*$' marker) in Figure 3.3 is traced out in a clockwise manner for increasing μ. The MSE ratio and the corresponding lower bound (3.140) are shown in Figure 3.4b.

3.5.2 Miss Probability

The blind estimator

$$\check{P}_M = \frac{1}{K\mathbb{P}\{\mathsf{u}=-1\}} \sum_{k=1}^{K} \frac{s\big(L_{\mathsf{u}}(\boldsymbol{x}_k) - \gamma\big)}{1 + e^{L_{\mathsf{u}}(\boldsymbol{x}_k)}} \tag{3.144}$$

is unbiased (cf. (3.69)) and its MSE is given by

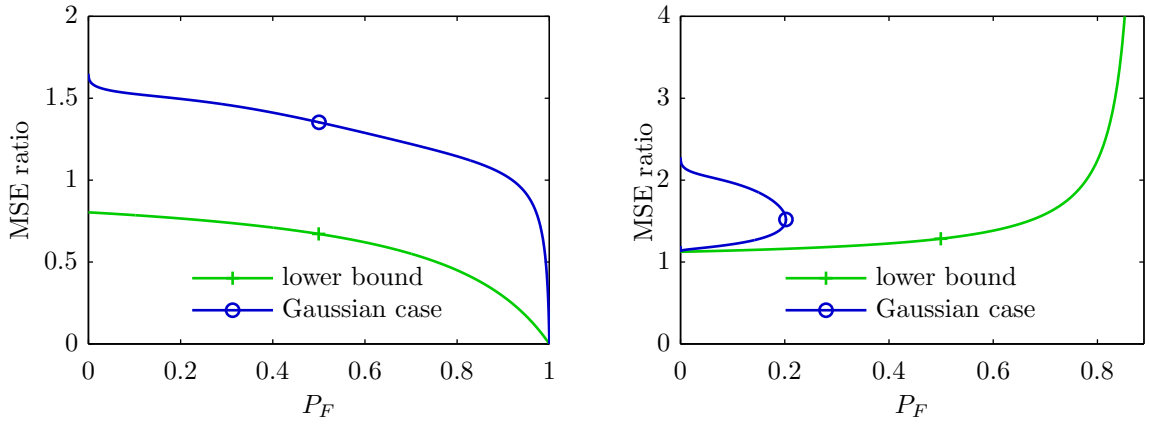
$$\mathrm{MSE}_{\check{P}_M}(P_M) = \mathbb{E}\{(\check{P}_M - P_M)^2\} = \frac{1}{K}\left(\frac{1}{\mathbb{P}\{\mathsf{u}=-1\}}\right)^2 \mathbb{E}\left\{\left(\frac{s(\mathsf{L}_{\mathsf{u}} - \gamma)}{1 + e^{\mathsf{L}_{\mathsf{u}}}}\right)^2\right\} - \frac{1}{K}P_M^2. \tag{3.145}$$

The expectation in (3.145) depends on the distribution of the data and in many cases (3.145) cannot be computed in closed form.

$$\left(\frac{s(\mathsf{L}_{\mathsf{u}} - \gamma)}{1 + e^{\mathsf{L}_{\mathsf{u}}}}\right)^2 \leq \frac{1}{1 + e^{\gamma}} \frac{s(\mathsf{L}_{\mathsf{u}} - \gamma)}{1 + e^{\mathsf{L}_{\mathsf{u}}}}. \tag{3.146}$$

The inequality in (3.146) yields the following sharp upper bound for the MSE:

$$\mathrm{MSE}_{\check{P}_M}(P_M) \leq \frac{P_M}{K}\left(\frac{1}{\mathbb{P}\{\mathsf{u}=-1\}(1 + e^{\gamma})} - P_M\right). \tag{3.147}$$



(a) $\mathbb{P}\{\mathsf{u}=1\} = 1/2$, $\gamma = 1/2$.

(b) $\mathbb{P}\{\mathsf{u}=1\} = 0.7$, $\gamma = 1/2$.

Figure 3.4: MSE ratio versus P_F. Comparison of $\mathrm{MSE}_{\hat{P}_F}(P_F)/\mathrm{MSE}_{\check{P}_F}(P_F)$ in the Gaussian case to the lower bound (3.140).

We have equality in (3.147) for any distribution of L_u that takes values only in $[-\infty, \gamma] \cup \{\infty\}$. However, for a specific distribution of the data, the upper bound in (3.147) may be rather loose. We note that the right-hand side of (3.147) is always nonnegative due to (3.70).

The nonblind estimator $\hat{P}_M = 1 - \hat{P}_D$ (cf. (3.66)) is unbiased and its MSE equals

$$\mathrm{MSE}_{\hat{P}_M}(P_M) = \frac{P_M}{\kappa}(1 - P_M), \tag{3.148}$$

where $\kappa = \sum_{k=1}^{K} \mathbb{1}\{u_k = 1\}$. The ratio of the MSEs in (3.148) and (3.145) is lower bounded as

$$\frac{\mathrm{MSE}_{\hat{P}_M}(P_M)}{\mathrm{MSE}_{\check{P}_M}(P_M)} \geq \frac{1 - P_M}{\left(\mathbb{P}\{\mathsf{u}=-1\}(1+e^{\gamma})\right)^{-1} - P_M}. \tag{3.149}$$

The following result is a consequence of the MSE ratio in (3.149).

Proposition 3.6. *The blind estimator $\check{P}_M$ dominates the nonblind estimator $\hat{P}_M$ for any distribution of the data if and only if*

$$\mathbb{P}\{\mathsf{u}=-1\}(1+e^{\gamma}) \geq 1. \tag{3.150}$$

For specific distributions of the data, $\check{P}_M$ may dominate $\hat{P}_M$ even if (3.150) does not hold.

Proof: The lower bound (3.149) shows that $\mathrm{MSE}_{\check{P}_M}(P_M) \leq \mathrm{MSE}_{\hat{P}_M}(P_M)$ for all P_M if the inequality (3.150) is satisfied. ■

Corollary 3.7. *Rewriting (3.150) in terms of $\mathbb{P}\{\mathsf{u}=1\}$ yields*

$$\mathbb{P}\{\mathsf{u}=1\}(1+e^{-\gamma}) \leq 1. \tag{3.151}$$

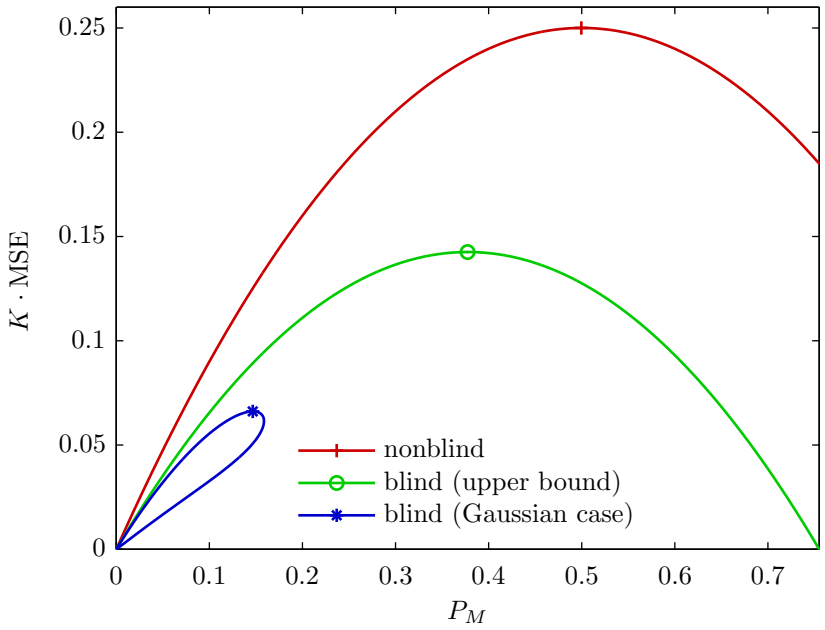


Figure 3.5: MSE versus P_M for $\mathbb{P}\{\mathsf{u}=1\} = 1/2$ and $\gamma = 1/2$. Comparison of $\mathrm{MSE}_{\check{P}_M}(P_M)$ (cf. (3.148)), upper bound (3.147), and $\mathrm{MSE}_{\hat{P}_M}(P_M)$ (cf. (3.145)) in the Gaussian case.

Comparing (3.151) and (3.141) shows that $\check{P}_M$ and $\check{P}_F$ simultaneously dominate $\hat{P}_M$ and $\hat{P}_F$, respectively, for any distribution of the data if and only if

$$\mathbb{P}\{\mathsf{u}=1\}(1+e^{-\gamma}) = 1. \tag{3.152}$$

Otherwise, either $\check{P}_M$ or $\check{P}_F$ dominates its corresponding nonblind estimator.

We next consider the same example as for the false alarm probability with conditionally Gaussian LLR, $\mathbb{P}\{\mathsf{u}=1\} = 1/2$, and $\gamma = 1/2$. In this case, the miss probability equals

$$P_M = Q\left(\frac{\gamma+\mu}{\sqrt{2\mu}}\right). \tag{3.153}$$

Here, (3.150) is fulfilled and therefore the blind estimator dominates the nonblind estimator. In Figure 3.5, we compare the MSE of the nonblind estimator (with $\kappa = K$) to the MSE of the blind estimator and the upper bound (3.147). In the Gaussian case, almost every P_M corresponds to two different values of μ, where the larger of the two values yields a smaller MSE. Hence, the blue curve ('$*$' marker) in Figure 3.5 is traced out in a clockwise manner for increasing μ. The MSE ratio and the corresponding lower bound (3.149) are shown in Figure 3.7a.

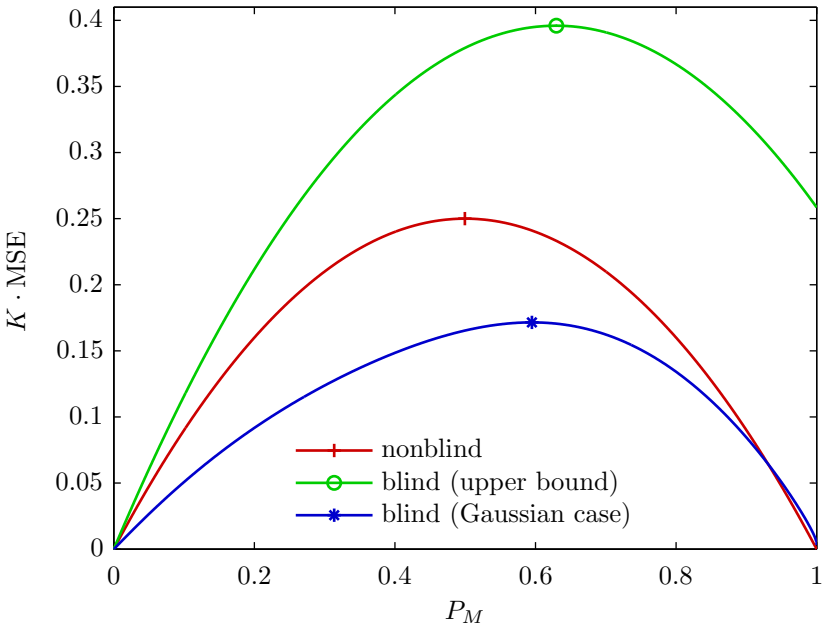


Figure 3.6: MSE versus P_M for $\mathbb{P}\{\mathsf{u}=1\} = 0.7$ and $\gamma = 1/2$. Comparison of $\mathrm{MSE}_{\hat{P}_M}(P_M)$ (cf. (3.148)), upper bound (3.147), and $\mathrm{MSE}_{\check{P}_M}(P_M)$ (cf. (3.145)) in the Gaussian case.

Next, we let $\mathbb{P}\{\mathsf{u}=1\} = 0.7$. The miss probability is then given by

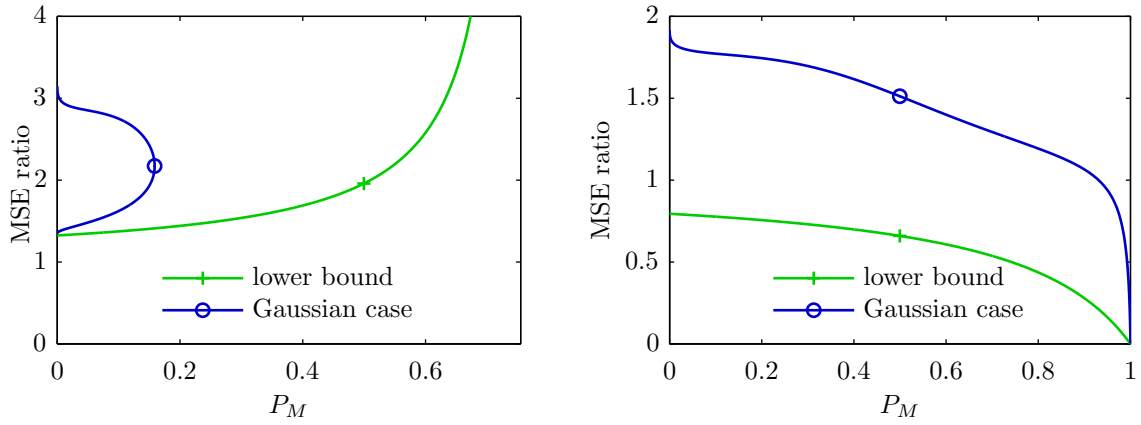
$$P_M = Q\left(\frac{\gamma + \mu - L_{\mathsf{u}}^a}{\sqrt{2\mu}}\right). \tag{3.154}$$

In Figure 3.6, we compare the MSE of the nonblind estimator (with $\kappa = K$) to the MSE of the blind estimator and the upper bound (3.147). Here, the upper bound (3.147) is greater than (3.148) for all P_M since (3.150) does not hold. However, the MSE of the blind estimator is smaller than the MSE of the nonblind estimator for all $P_M \lesssim 0.93$, i.e., the blind estimator outperforms the nonblind estimator for all P_M of practical interest. The MSE ratio and the corresponding lower bound (3.149) are shown in Figure 3.4b.

3.5.3 Detection Probability and Acceptance Probability

The detection probability equals $P_D = 1 - P_M$ and thus $\check{P}_D = 1 - \check{P}_M$, $\hat{P}_D = 1 - \hat{P}_M$. The MSE of $\check{P}_D$ is given by

$$\begin{aligned}
\mathrm{MSE}_{\check{P}_D}(P_D) &= \mathbb{E}\big\{\big(\check{P}_M - (1 - P_D)\big)^2\big\} && (3.155)\\
&= \frac{1}{K}\left(\frac{1}{\mathbb{P}\{\mathsf{u}=-1\}}\right)^2 \mathbb{E}\left\{\left(\frac{s(\mathsf{L}_{\mathsf{u}} - \gamma)}{1 + e^{\mathsf{L}_{\mathsf{u}}}}\right)^2\right\} - \frac{1}{K}(1 - P_D)^2 && (3.156)\\
&\leq \frac{1 - P_D}{K}\left(\frac{1}{\mathbb{P}\{\mathsf{u}=-1\}(1 + e^{\gamma})} - (1 - P_D)\right). && (3.157)
\end{aligned}$$



(a) $\mathbb{P}\{\mathsf{u}=1\} = 1/2$, $\gamma = 1/2$.

(b) $\mathbb{P}\{\mathsf{u}=1\} = 0.7$, $\gamma = 1/2$.

Figure 3.7: MSE ratio versus P_M. Comparison of $\mathrm{MSE}_{\hat{P}_M}(P_M)/\mathrm{MSE}_{\check{P}_M}(P_M)$ in the Gaussian case to the lower bound (3.149).

The MSE of $\hat{P}_D$ equals

$$\mathrm{MSE}_{\hat{P}_D}(P_D) = \frac{P_D}{\kappa}(1-P_D). \tag{3.158}$$

The estimators $\hat{P}_D$ and $\check{P}_D$ are both unbiased and the ratio of their MSEs is lower bounded as

$$\frac{\mathrm{MSE}_{\hat{P}_D}(P_D)}{\mathrm{MSE}_{\check{P}_D}(P_D)} \geq \frac{P_D}{\left(\mathbb{P}\{\mathsf{u}=-1\}(1+e^{\gamma})\right)^{-1} - (1-P_D)}. \tag{3.159}$$

Clearly, $\check{P}_D$ dominates $\hat{P}_D$ if and only if $\check{P}_M$ dominates $\hat{P}_M$.

Similarly, for the acceptance probability P_A we have

$$\begin{aligned}
\mathrm{MSE}_{\check{P}_A}(P_A) &= \mathbb{E}\left\{\left(\check{P}_F - (1-P_A)\right)^2\right\} && (3.160)\\
&= \frac{1}{K}\left(\frac{1}{\mathbb{P}\{\mathsf{u}=1\}}\right)^2 \mathbb{E}\left\{\left(\frac{s(\gamma-\mathsf{L}_\mathsf{u})}{1+e^{-\mathsf{L}_\mathsf{u}}}\right)^2\right\} - \frac{1}{K}(1-P_A)^2 && (3.161)\\
&\leq \frac{1-P_A}{K}\left(\frac{1}{\mathbb{P}\{\mathsf{u}=1\}(1+e^{-\gamma})} - (1-P_A)\right) && (3.162)
\end{aligned}$$

and

$$\mathrm{MSE}_{\hat{P}_A}(P_A) = \frac{P_A}{\kappa}(1-P_A). \tag{3.163}$$

The estimators $\hat{P}_A$ and $\check{P}_A$ are both unbiased and the ratio of their MSEs is lower bounded as

$$\frac{\mathrm{MSE}_{\hat{P}_A}(P_A)}{\mathrm{MSE}_{\check{P}_A}(P_A)} \geq \frac{P_A}{\left(\mathbb{P}\{\mathsf{u}=1\}(1+e^{-\gamma})\right)^{-1} - (1-P_A)}. \tag{3.164}$$

Again, $\check{P}_A$ dominates $\hat{P}_A$ if and only if $\check{P}_F$ dominates $\hat{P}_F$.

3.5.4 Conditional Error Probability

The blind estimator

$$\check{P}_e(u) = \frac{1}{P\{\mathsf{u}=u\}} \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\{\hat{u}(\boldsymbol{x}_k) \neq u\} \mathbb{P}\{\mathsf{u}=u|\mathsf{x}=\boldsymbol{x}_k\}, \quad u \in \mathcal{U}, \tag{3.165}$$

for the conditional error probability given $\mathsf{u} = u$ is unbiased (cf. (3.76)) and its MSE equals

$$\begin{aligned} \mathrm{MSE}_{\check{P}_e(u)}\big(P_e(u)\big) &= \mathbb{E}\big\{\big(\check{P}_e(u) - P_e(u)\big)^2\big\} & (3.166)\\ &= \frac{1}{K}\left(\frac{1}{\mathbb{P}\{\mathsf{u}=u\}}\right)^2 \mathbb{E}\big\{\,(\mathbb{1}\{\hat{u}(\mathsf{x}) \neq u\}\mathbb{P}\{\mathsf{u}=u|\mathsf{x}\}\big)^2\,\big\} - \frac{1}{K} P_e(u)^2. & (3.167) \end{aligned}$$

Since the term in the expectation in (3.167) is at most equal to 1, we can bound the MSE as follows:

$$\mathrm{MSE}_{\check{P}_e(u)}\big(P_e(u)\big) \leq \frac{P_e(u)}{K}\left(\frac{1}{\mathbb{P}\{\mathsf{u}=u\}} - P_e(u)\right), \quad u \in \mathcal{U}. \tag{3.168}$$

In contrast to (3.167), the upper bound in (3.168) does not depend on the actual distribution of the data.

The nonblind estimator

$$\hat{P}_e(u) = \frac{1}{\sum_{k=1}^{K} \mathbb{1}\{u_k=u\}} \sum_{k=1}^{K} \mathbb{1}\{\hat{u}(\boldsymbol{x}_k) \neq u\}\mathbb{1}\{u_k=u\}, \quad u \in \mathcal{U}, \tag{3.169}$$

is unbiased and its MSE equals

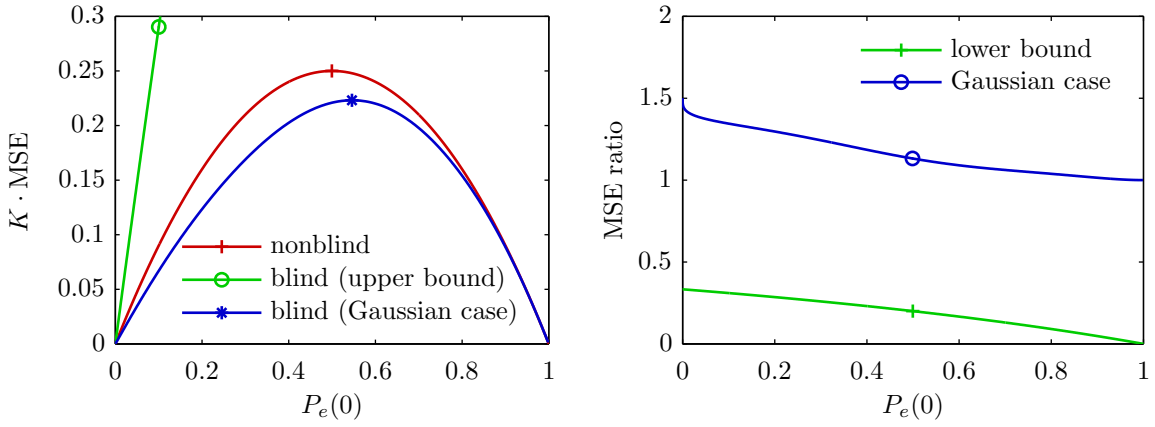
$$\mathrm{MSE}_{\hat{P}_e(u)}\big(P_e(u)\big) = \frac{P_e(u)}{\kappa}\big(1 - P_e(u)\big), \quad u \in \mathcal{U}, \tag{3.170}$$

where $\kappa = \sum_{k=1}^{K} \mathbb{1}\{u_k = u\}$ denotes the number of source outputs which are equal to u. Of course, the MSE in (3.170) is smallest if $\kappa = K$ which can be achieved by choosing $u_1 = \ldots = u_K = u$ (if the source outputs are training data). The ratio of the MSEs in (3.170) and (3.167) is lower bounded as

$$\frac{\mathrm{MSE}_{\hat{P}_e(u)}\big(P_e(u)\big)}{\mathrm{MSE}_{\check{P}_e(u)}\big(P_e(u)\big)} \geq \frac{1 - P_e(u)}{(\mathbb{P}\{\mathsf{u}=u\})^{-1} - P_e(u)}. \tag{3.171}$$

The bound in (3.171) tells us that $\check{P}_e(u)$ dominates $\hat{P}_e(u)$ for any distribution of the data only in the trivial case where $\mathbb{P}\{\mathsf{u}=u\} = 1$. This is in contrast to the binary case, where we can tighten the upper bound (3.168) for the MSE using the bounds on the conditional error probabilities P_F and P_M. For specific distributions of the data, the blind estimator $\check{P}_e(u)$ may nevertheless dominate $\hat{P}_e(u)$ as the following example shows.

We assume $\mathcal{U} = \{-1, 0, 1\}$ and equally likely $\mathsf{u} \in \mathcal{U}$. Furthermore, let $\mathsf{x} = \mathsf{u} + \mathsf{w}$ with $\mathsf{w} \sim \mathcal{N}(0, \sigma^2)$. We assume that a MAP detector is used, i.e., $\hat{u}(x) = \arg\min_{u \in \mathcal{U}} |x - u|$. For



(a) (b)

Figure 3.8: MAP detection of a 3-ary signal in Gaussian noise. (a) Comparison of $\mathrm{MSE}_{\hat{P}_e(u)}\big(P_e(0)\big)$ (cf. (3.170)), upper bound (3.168), and $\mathrm{MSE}_{\check{P}_e(u)}\big(P_e(0)\big)$ (cf. (3.167)) in the Gaussian case. (b) Comparison of $\mathrm{MSE}_{\hat{P}_e(u)}\big(P_e(0)\big)/\mathrm{MSE}_{\check{P}_e(u)}\big(P_e(0)\big)$ in the Gaussian case to the lower bound (3.171).

the conditional error probability $P_e(0)$ we then have

$$P_e(0) = 2Q\left(\frac{1}{2\sigma}\right). \tag{3.172}$$

In Figure 3.8a, we plot the MSE of the nonblind estimator (for $\kappa = K$), the MSE of the blind estimator, and the upper bound (3.168). We observe that the blind estimator dominates the nonblind estimator. Furthermore, the upper bound (3.168) is loose in this case. The MSE ratio in Figure 3.8b shows that for small $P_e(0)$, the blind estimator outperforms the corresponding nonblind estimator by a factor of approximately 1.4.

3.5.5 Error Probability

The blind estimator

$$\check{P}_e = 1 - \frac{1}{K}\sum_{k=1}^{K} \mathbb{P}\{\mathsf{u}=\hat{u}(\mathsf{x})|\mathsf{x}=\boldsymbol{x}_k\} \tag{3.173}$$

for the (unconditional) error probability is unbiased (cf. (3.81)). The MSE of $\check{P}_e$ is given by

$$\mathrm{MSE}_{\check{P}_e}(P_e) = \mathbb{E}\big\{(\check{P}_e - P_e)^2\big\} = \frac{1}{K}\mathbb{E}\big\{(1 - \mathbb{P}\{\mathsf{u}=\hat{u}(\mathsf{x})|\mathsf{x}\})^2\big\} - \frac{1}{K}P_e^2. \tag{3.174}$$

Depending on the distribution of the data, the expectation in (3.174) may be hard to compute in closed form. However, we can upper bound the MSE by noting that

$$\mathbb{E}\big\{(1 - \mathbb{P}\{\mathsf{u}=\hat{u}(\mathsf{x})|\mathsf{x}\})^2\big\} \leq \mathbb{E}\big\{1 - \mathbb{P}\{\mathsf{u}=\hat{u}(\mathsf{x})|\mathsf{x}\}\big\} = P_e. \tag{3.175}$$

Using (3.175) in (3.174) yields

$$\mathrm{MSE}_{\check{P}_e}(P_e) \leq \frac{P_e}{K}(1-P_e). \tag{3.176}$$

The MSE of the unbiased nonblind estimator $\hat{P}_e = \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\{\hat{u}(\boldsymbol{x}_k) \neq u_k\}$ equals

$$\mathrm{MSE}_{\hat{P}_e}(P_e) = \frac{P_e}{K}(1-P_e), \tag{3.177}$$

and thus the ratio of the MSEs in (3.177) and (3.174) is lower bounded as

$$\frac{\mathrm{MSE}_{\hat{P}_e}(P_e)}{\mathrm{MSE}_{\check{P}_e}(P_e)} \geq 1. \tag{3.178}$$

The inequality (3.178) allows us to state the following result.

Proposition 3.8. *The blind estimator $\check{P}_e$ dominates the nonblind estimator $\hat{P}_e$ for any distribution of the data.*

Proof: The proposition follows directly from (3.178). ■

To illustrate these results, we use the same example (3-ary signal in Gaussian noise) as for the conditional error probability. The error probability of the MAP detector is then given by

$$P_e = \frac{4}{3}Q\left(\frac{1}{2\sigma}\right) \leq \frac{2}{3}. \tag{3.179}$$

In Figure 3.9a, we plot the MSE of the nonblind estimator (which equals the upper bound of the blind estimator's MSE) and the MSE of the blind estimator. We observe that the blind estimator significantly outperforms the nonblind estimator for all P_e. Figure 3.9b shows that the MSE ratio $\mathrm{MSE}_{\hat{P}_e}(P_e)/\mathrm{MSE}_{\check{P}_e}(P_e)$ is greater than 4 for small values of P_e and it tends to infinity as $P_e \to 2/3$.

In the binary case, the upper bound in (3.176) can be tightened. Specifically, we can rewrite the MSE (3.174) as follows:

$$\mathrm{MSE}_{\check{P}_e}(P_e) = \frac{1}{K}\mathbb{E}\left\{\left(\frac{s(\gamma - L_{\mathsf{u}}(\mathbf{x}))}{1+e^{-L_{\mathsf{u}}(\mathbf{x})}} + \frac{s(L_{\mathsf{u}}(\mathbf{x})-\gamma)}{1+e^{L_{\mathsf{u}}(\mathbf{x})}}\right)^2\right\} - \frac{1}{K}P_e^2. \tag{3.180}$$

We note that

$$\frac{s(\gamma - L_{\mathsf{u}}(\mathbf{x}))}{1+e^{-L_{\mathsf{u}}(\mathbf{x})}} + \frac{s(L_{\mathsf{u}}(\mathbf{x})-\gamma)}{1+e^{L_{\mathsf{u}}(\mathbf{x})}} \leq \frac{1}{1+e^{-|\gamma|}}, \tag{3.181}$$

which allows us to upper bound (3.180) as

$$\mathrm{MSE}_{\check{P}_e}(P_e) \leq \frac{P_e}{K}\left(\frac{1}{1+e^{-|\gamma|}} - P_e\right). \tag{3.182}$$

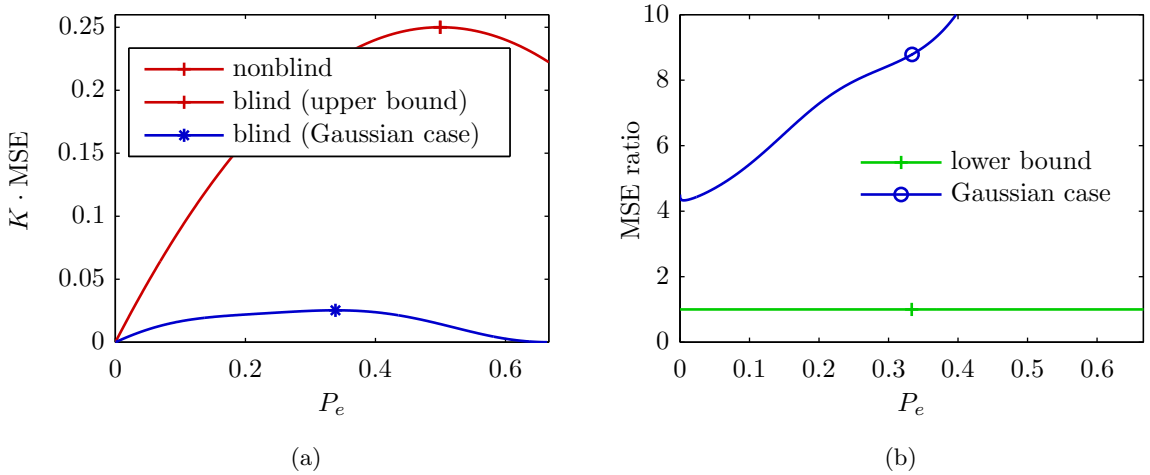


Figure 3.9: MAP detection of a 3-ary signal in Gaussian noise. (a) Comparison of $\mathrm{MSE}_{\hat{P}_e}(P_e)$ (cf. (3.170)), upper bound (3.168), and $\mathrm{MSE}_{\check{P}_e}(P_e)$ (cf. (3.167)) in the Gaussian case. (b) Comparison of $\mathrm{MSE}_{\hat{P}_e}(P_e)/\mathrm{MSE}_{\check{P}_e}(P_e)$ in the Gaussian case to the lower bound (3.171).

Hence, the MSE ratio $\mathrm{MSE}_{\hat{P}_e}(P_e)/\mathrm{MSE}_{\check{P}_e}(P_e)$ is lower bounded as follows:

$$\frac{\mathrm{MSE}_{\hat{P}_e}(P_e)}{\mathrm{MSE}_{\check{P}_e}(P_e)} \geq \frac{1-P_e}{(1+e^{-|\gamma|})^{-1}-P_e}. \tag{3.183}$$

We note that (3.90) ensures that the denominator in (3.183) is nonnegative. Furthermore, the bounds in (3.182) and (3.183) cannot be tightened, i.e., the inequalities are sharp. Indeed, any distribution of the data such that $L_{\mathsf{u}}(\mathbf{x}) \in \{\gamma, \pm\infty\}$ achieves equality in (3.182) and (3.183). The following result is a consequence of (3.183).

Proposition 3.9. *In the binary case, the blind estimator $\check{P}_e$ outperforms the nonblind estimator $\hat{P}_e$ for any distribution of the data by at least a factor of $1+e^{-|\gamma|}$, i.e., we have*

$$\frac{MSE_{\hat{P}_e}(P_e)}{MSE_{\check{P}_e}(P_e)} \geq 1+e^{-|\gamma|}. \tag{3.184}$$

Proof: The lower bound (3.183) is minimal when $P_e = 0$. The minimum value equals $1+e^{-|\gamma|} \in [1,2]$. ∎

Due to Proposition 3.9, $\check{P}_e$ outperforms $\hat{P}_e$ at least by a factor of 2 for the special case of a MAP detector. In this case, the MSE of $\check{P}_e$ is given by

$$\mathrm{MSE}_{\check{P}_e}(P_e) = \frac{1}{K}\mathbb{E}\left\{\left(\frac{1}{1+e^{|L_{\mathsf{u}}(\mathbf{x})|}}\right)^2\right\} - \frac{1}{K}P_e^2 \tag{3.185}$$

$$\leq \frac{P_e}{K}(1/2 - P_e). \tag{3.186}$$

The following results provides a convenient alternative to numerical integration for the evaluation of the expectation in (3.185) using the absolute moments of L_{u}.

Proposition 3.10. *Assuming* $\mathsf{L_u}$ *has finite absolute moments, the mean power of* $(1+e^{|\mathsf{L_u}|})^{-1}$ *equals*

$$\mathbb{E}\left\{\left(\frac{1}{1+e^{|\mathsf{L_u}|}}\right)^2\right\} = \frac{1}{4} + \sum_{m=1}^{\infty} \frac{\mathbb{E}\{|\mathsf{L_u}|^m\}}{m!} \sum_{k=1}^{m} \frac{(-1)^k}{2^{k+2}} d_{k,m}, \tag{3.187}$$

where the coefficients $d_{k,m}$ *are defined by the recursion*

$$d_{k,m} = (k+1)d_{k-1,m-1} + k d_{k,m-1} \quad \textit{for } k \geq 2, m \geq 2, \tag{3.188a}$$

$$d_{1,m} = 2 \quad \textit{for } m \geq 1, \tag{3.188b}$$

$$d_{k,1} = 0 \quad \textit{for } k \geq 2. \tag{3.188c}$$

Proof: See Appendix A.2. ∎

The sign of the terms in the series (3.187) can be shown to change after every second term. Therefore, we can truncate the series after any pair of terms having the same sign and bound the error by the sum of the following two terms. We can further expand $\mathbb{E}\{|\mathsf{L_u}|^m\}$ in (3.187) as

$$\mathbb{E}\{|\mathsf{L_u}|^m\} = \mathbb{E}\{|\mathsf{L_u}|^m|\mathsf{u}{=}1\}\mathbb{P}\{\mathsf{u}{=}1\} + \mathbb{E}\{|\mathsf{L_u}|^m|\mathsf{u}{=}{-1}\}\mathbb{P}\{\mathsf{u}{=}{-1}\}. \tag{3.189}$$

For conditionally Gaussian LLRs with $\mathsf{L_u}|u \sim \mathcal{N}(u\mu, 2\mu)$, $\mu > 0$, we have

$$\mathbb{E}\{|\mathsf{L_u}|^m|\mathsf{u}{=}u\} = \frac{2^m|\mu|^{m/2}}{\sqrt{\pi}}\Gamma\left(\frac{m+1}{2}\right)\Phi\left(-\frac{m}{2}, \frac{1}{2}; -\frac{|\mu|}{4}\right), \tag{3.190}$$

where $\Gamma(\cdot)$ and $\Phi(\cdot,\cdot;\cdot)$ respectively denote the gamma function and Kummer's confluent hypergeometric function (cf. Appendix D for details). We note that (3.190) does not depend on u, i.e., we have $\mathbb{E}\{|\mathsf{L_u}|^m|\mathsf{u} = u\} = \mathbb{E}\{|\mathsf{L_u}|^m\}$, since $|\mathsf{L_u}|^m$ is an even function and the distribution of $\mathsf{L_u}$ is even (cf. (3.38)).

A simpler way to compute the MSE of the blind estimator in the binary case with MAP detection is via the moments of the soft bit $\mathsf{\Lambda_u}$. Rewriting (3.185) in terms of $\mathsf{\Lambda_u}$ yields

$$\mathrm{MSE}_{\check{P}_e}(P_e) = \frac{1}{4K}\mathbb{E}\{(1-|\mathsf{\Lambda_u}|)^2\} - \frac{1}{K}P_e^2 = \frac{1}{K}P_e(1-P_e) - \frac{1}{4K}(1-\mathbb{E}\{\mathsf{\Lambda_u}^2\}) \tag{3.191}$$

$$= \mathrm{MSE}_{\hat{P}_e}(P_e) - \frac{1}{4K}\varepsilon_{\mathrm{MMSE}}. \tag{3.192}$$

Using (3.191), we only require the mean power of $\mathsf{\Lambda_u}$ to compute $\mathrm{MSE}_{\check{P}_e}(P_e)$. Further rewriting (3.191) as in (3.192) yields an interesting connection between the MSEs of the blind and nonblind estimators, and the minimum MSE, respectively. Specifically, $\varepsilon_{\mathrm{MMSE}}/(4K)$ can be viewed as the penalty for using hard decisions in the estimation of P_e.

For the binary case with MAP detection, Figure 3.10a shows the MSE of the nonblind estimator as well as the MSE of the blind estimator and the upper bound (3.182). As in

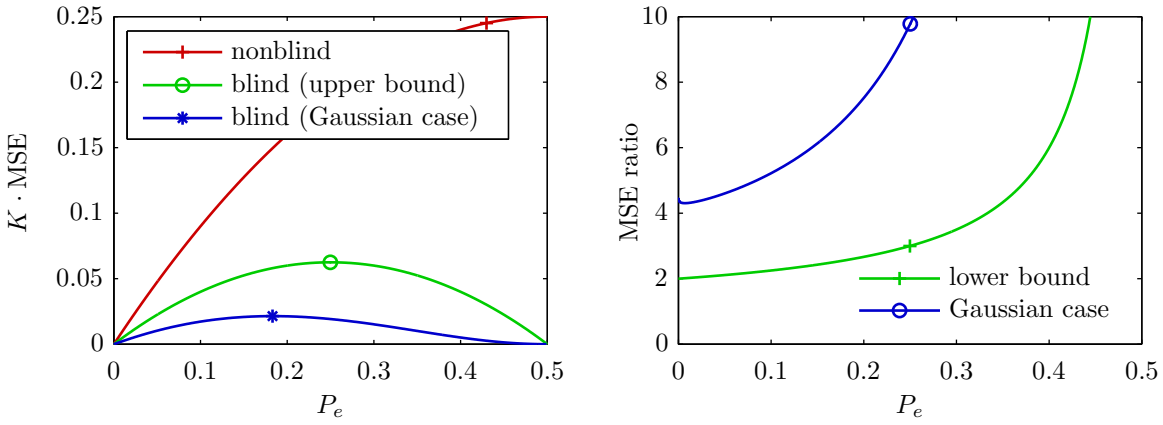


Figure 3.10: The binary case with MAP detection. (a) Comparison of $\mathrm{MSE}_{\hat{P}_e}(P_e)$ (cf. (3.170)), upper bound (3.186), and $\mathrm{MSE}_{\check{P}_e}(P_e)$ (cf. (3.185)) in the Gaussian case. (b) Comparison of $\mathrm{MSE}_{\hat{P}_e}(P_e)/\mathrm{MSE}_{\check{P}_e}(P_e)$ in the Gaussian case to the lower bound (3.184).

the nonbinary case, the blind estimator significantly outperforms the nonblind estimator for all P_e. Figure 3.10b shows that the MSE ratio $\mathrm{MSE}_{\hat{P}_e}(P_e)/\mathrm{MSE}_{\check{P}_e}(P_e)$ is greater than 4 for small values of P_e. Both, the lower bound (3.183) and the true MSE ratio tend to infinity as $P_e \to 1/2$.

3.5.6 Block Error Probability

To simplify the MSE analysis for the block error probability, we assume that all error events are statistically independent. In this case, the block error probability for a length-N block equals

$$P_b = 1 - \prod_{n=1}^{N} (1 - P_{e,n}), \tag{3.193}$$

where $P_{e,n}$ denotes the error probability of the nth source output in the block. Due to the independence of the error events, the blind estimator

$$\check{P}_b = 1 - \prod_{n=1}^{N} \frac{1}{K} \sum_{k=1}^{K} \mathbb{P}\{\mathsf{u}_n = \hat{u}(\mathsf{x}_n) | \mathsf{x}_n = \boldsymbol{x}_{n,k}\} \tag{3.194}$$

is unbiased. Here, $\boldsymbol{x}_{n,k}$ denotes the kth observation of the data corresponding to the nth source output. The MSE of $\check{P}_b$ is given by

$$\mathrm{MSE}_{\check{P}_b}(P_b) = \mathbb{E}\big\{(\check{P}_b - P_b)^2\big\} \tag{3.195}$$

$$= \mathbb{E}\left\{ \left(1 - \prod_{n=1}^{N} \frac{1}{K} \sum_{k=1}^{K} \mathbb{P}\{\mathsf{u}_n = \hat{u}(\mathsf{x}_n) | \mathsf{x}_n = \mathsf{x}_{n,k}\} \right)^2 \right\} - \left(1 - \prod_{n=1}^{N} (1 - P_{e,n}) \right)^2 \tag{3.196}$$

$$= \prod_{n=1}^{N} \left(\frac{1}{K} \mathbb{E}\big\{ (\mathbb{P}\{\mathsf{u}_n = \hat{u}(\mathsf{x}_n)|\mathsf{x}_n\})^2 \big\} + (1 - 1/K)(1 - P_{e,n})^2 \right) - \prod_{n=1}^{N} (1 - P_{e,n})^2 \tag{3.197}$$

$$= \prod_{n=1}^{N} \left(\frac{1}{K} \mathbb{E}\big\{ (\mathbb{P}\{\mathsf{u}_n \neq \hat{u}(\mathsf{x}_n)|\mathsf{x}_n\})^2 \big\} - \frac{1}{K} P_{e,n}^2 + (1 - P_{e,n})^2 \right) - \prod_{n=1}^{N} (1 - P_{e,n})^2. \tag{3.198}$$

Similarly, the blind estimator

$$\check{\check{P}}_b = 1 - \frac{1}{K} \sum_{k=1}^{K} \prod_{n=1}^{N} \mathbb{P}\{\mathsf{u}_n = \hat{u}(\mathsf{x}_n)|\mathsf{x}_n = \boldsymbol{x}_{n,k}\} \tag{3.199}$$

is unbiased and its MSE equals

$$\mathrm{MSE}_{\check{\check{P}}_b}(P_b) = \mathbb{E}\big\{ (\check{P}_b - P_b)^2 \big\} \tag{3.200}$$

$$= \frac{1}{K} \prod_{n=1}^{N} \mathbb{E}\big\{ (\mathbb{P}\{\mathsf{u}_n = \hat{u}(\mathsf{x}_n)|\mathsf{x}_n\})^2 \big\} - \frac{1}{K} \prod_{n=1}^{N} (1 - P_{e,n})^2. \tag{3.201}$$

Note that the blind estimators (3.194) and (3.199) are equal if $K = 1$. To bound the MSEs (3.198) and (3.201), we note that

$$(\mathbb{P}\{\mathsf{u}_n \neq \hat{u}(\mathsf{x}_n)|\mathsf{x}_n = \boldsymbol{x}_n\})^2 \leq \alpha_n \mathbb{P}\{\mathsf{u}_n \neq \hat{u}(\mathsf{x}_n)|\mathsf{x}_n = \boldsymbol{x}_n\}, \tag{3.202}$$

for all $\boldsymbol{x}_n \in \mathcal{X}^{(n)}$ and some $\alpha_n \in [0, 1)$ (here, $\mathcal{X}^{(n)}$ denotes the observation space corresponding to the nth source output). More specifically, we choose $\alpha_n = \arg\min_{\beta_n} f(\beta_n)$, where $f(\beta_n) = \beta_n \mathbb{P}\{\mathsf{u}_n \neq \hat{u}(\mathsf{x}_n)|\mathsf{x}_n = \boldsymbol{x}_n\} - (\mathbb{P}\{\mathsf{u}_n \neq \hat{u}(\mathsf{x}_n)|\mathsf{x}_n = \boldsymbol{x}_n\})^2$, subject to the constraint $f(\beta_n) \geq 0$ for all $\boldsymbol{x}_n \in \mathcal{X}^{(n)}$ and any distribution of the data. In the binary case, α_n is given by the right-hand side of (3.89). In the nonbinary case, we can not give a general closed-form expression for α_n.

Using (3.202) we upper bound (3.198) and (3.201) as follows:

$$\mathrm{MSE}_{\check{P}_b}(P_b) \leq \prod_{n=1}^{N} \big(1 - (2 - \alpha_n/K) P_{e,n} + (1 - 1/K) P_{e,n}^2\big) - \prod_{n=1}^{N} (1 - P_{e,n})^2 \tag{3.203}$$

$$\leq \frac{1}{K} \prod_{n=1}^{N} (1 - (2 - \alpha_n) P_{e,n}) - \frac{1}{K} \prod_{n=1}^{N} (1 - P_{e,n})^2, \tag{3.204}$$

and

$$\mathrm{MSE}_{\check{\check{P}}_b}(P_b) \leq \frac{1}{K} \prod_{n=1}^{N} (1 - (2 - \alpha_n) P_{e,n}) - \frac{1}{K} \prod_{n=1}^{N} (1 - P_{e,n})^2. \tag{3.205}$$

We note that (3.203) and (3.204) are equal for $K = 1$ and the weaker upper bound (3.204) for $\check{P}_b$ equals the upper bound (3.205) for $\check{\check{P}}_b$. A proof of (3.204) is given in Appendix A.3.

In the above expressions, the block error probability P_b is determined by $P_{e,n}$, $n = 1, \ldots, N$, according to (3.193). For the special case where the data model is equal for each source output, we have $P_{e,n} \equiv P_e$ and $\alpha_n \equiv \alpha$. In turn, this entails $P_b = 1 - (1 - P_e)^N$, or, equivalently, $P_e = 1 - (1 - P_b)^{1/N}$. Letting additionally $N \to \infty$ in (3.203)-(3.205) yields

$$\lim_{N\to\infty} \mathrm{MSE}_{\check{P}_b}(P_b) \leq (1-P_b)^{2-\alpha/K} - (1-P_b)^2 \leq \frac{1}{K}\big((1-P_b)^{2-\alpha} - (1-P_b)^2\big), \tag{3.206}$$

$$\lim_{N\to\infty} \mathrm{MSE}_{\check{\check{P}}_b}(P_b) \leq \frac{1}{K}\big((1-P_b)^{2-\alpha} - (1-P_b)^2\big). \tag{3.207}$$

For finite N and large K we obtain the limit

$$\lim_{K\to\infty} K\,\mathrm{MSE}_{\check{P}_b}(P_b) \leq \lim_{K\to\infty} K\left(1 - (2-\alpha/K)P_e + (1-1/K)P_e^2\right)^N - K(1-P_e)^{2N} \tag{3.208}$$

$$= NP_e(\alpha - P_e)(1-P_e)^{2(N-1)} \tag{3.209}$$

$$\leq (1-(2-\alpha)P_e)^N - (1-P_e)^{2N}. \tag{3.210}$$

The asymptotic case where both N and K are large yields

$$\lim_{K\to\infty}\lim_{N\to\infty} K\,\mathrm{MSE}_{\check{P}_b}(P_b) \leq \lim_{K\to\infty} K(1-P_b)^{2-\alpha/K} - K(1-P_b)^2 \tag{3.211}$$

$$= \alpha(1-P_b)^2 \log\frac{1}{1-P_b} \tag{3.212}$$

$$\leq (1-P_b)^{2-\alpha} - (1-P_b)^2. \tag{3.213}$$

Clearly, the weaker bounds (3.210) and (3.213) are upper bounds for the corresponding limits of $K\,\mathrm{MSE}_{\check{\check{P}}_b}(P_b)$.

The nonblind estimator

$$\hat{P}_b = 1 - \frac{1}{K}\sum_{k=1}^{K}\prod_{n=1}^{N} \mathbb{1}\{\hat{u}(\boldsymbol{x}_{n,k}) = u_n\} \tag{3.214}$$

is unbiased and its MSE equals

$$\mathrm{MSE}_{\hat{P}_b}(P_b) = \frac{P_b}{K}(1-P_b) = \frac{1}{K}\prod_{n=1}^{N}(1-P_{e,n}) - \frac{1}{K}\prod_{n=1}^{N}(1-P_{e,n})^2. \tag{3.215}$$

The ratio of the MSEs in (3.215) and (3.195) is lower bounded as

$$\frac{\mathrm{MSE}_{\hat{P}_b}(P_b)}{\mathrm{MSE}_{\check{P}_b}(P_b)} \geq \frac{1}{K}\frac{\prod_{n=1}^{N}(1-P_{e,n}) - \prod_{n=1}^{N}(1-P_{e,n})^2}{\prod_{n=1}^{N}\left(1-(2-\alpha_n/K)P_{e,n} + (1-1/K)P_{e,n}^2\right) - \prod_{n=1}^{N}(1-P_{e,n})^2} \tag{3.216}$$

$$\geq \frac{\prod_{n=1}^{N}(1-P_{e,n}) - \prod_{n=1}^{N}(1-P_{e,n})^2}{\prod_{n=1}^{N}(1-(2-\alpha_n)P_{e,n}) - \prod_{n=1}^{N}(1-P_{e,n})^2}. \tag{3.217}$$

Similarly, the ratio of the MSEs in (3.215) and (3.200) is lower bounded as

$$\frac{\mathrm{MSE}_{\hat{P}_b}(P_b)}{\mathrm{MSE}_{\check{\check{P}}_b}(P_b)} \geq \frac{\prod_{n=1}^{N}(1-P_{e,n}) - \prod_{n=1}^{N}(1-P_{e,n})^2}{\prod_{n=1}^{N}(1-(2-\alpha_n)P_{e,n}) - \prod_{n=1}^{N}(1-P_{e,n})^2}. \tag{3.218}$$

The bounds (3.216)-(3.218) allow us to state the following result.

Proposition 3.11. *The blind estimators $\check{P}_b$ and $\check{\check{P}}_b$ dominate the nonblind estimator $\hat{P}_b$ for any distribution of the data by at least a factor of $\min_n \alpha_n^{-1}$, i.e., we have*

$$\frac{MSE_{\hat{P}_b}(P_b)}{MSE_{\check{P}_b}(P_b)} \geq \min_n \alpha_n^{-1}, \tag{3.219}$$

and

$$\frac{MSE_{\hat{P}_b}(P_b)}{MSE_{\check{\check{P}}_b}(P_b)} \geq \min_n \alpha_n^{-1}. \tag{3.220}$$

We have equality in (3.219) and (3.220) if and only if $P_{e,n} = 0$, $n = 1, \ldots, N$.

Proof: See Appendix A.4. ■

Due to Proposition 3.11, $\check{P}_b$ and $\check{\check{P}}_b$ outperform $\hat{P}_b$ in the binary case at least by a factor of 2 for the special case of a MAP detector (note that in this case $\alpha_n = 1/2$, $n = 1, \ldots, N$). For the case where $P_{e,n} \equiv P_e$, $\alpha_n \equiv \alpha$, and $N \to \infty$, the MSE ratio is lower bounded as follows:

$$\lim_{N\to\infty} \frac{\mathrm{MSE}_{\hat{P}_b}(P_b)}{\mathrm{MSE}_{\check{P}_b}(P_b)} \geq \frac{1}{K} \frac{P_b}{(1-P_b)^{1-\alpha/K} - (1-P_b)} \geq \frac{P_b}{(1-P_b)^{1-\alpha} - (1-P_b)}, \tag{3.221}$$

$$\lim_{N\to\infty} \frac{\mathrm{MSE}_{\hat{P}_b}(P_b)}{\mathrm{MSE}_{\check{\check{P}}_b}(P_b)} \geq \frac{P_b}{(1-P_b)^{1-\alpha} - (1-P_b)}. \tag{3.222}$$

In the limit of large K, the MSE ratio is lower bounded as

$$\lim_{K\to\infty} \frac{\mathrm{MSE}_{\hat{P}_b}(P_b)}{\mathrm{MSE}_{\check{P}_b}(P_b)} \geq \frac{(1-P_e)^N - (1-P_e)^{2N}}{NP_e(\alpha - P_e)(1-P_e)^{2(N-1)}} \tag{3.223}$$

$$\geq \frac{(1-P_e)^N - (1-P_e)^{2N}}{(1-(2-\alpha)P_e)^N (1-P_e)^{2N}}. \tag{3.224}$$

The MSE ratio in the limit of large N and K is lower bounded as follows:

$$\lim_{K\to\infty} \lim_{N\to\infty} \frac{\mathrm{MSE}_{\hat{P}_b}(P_b)}{\mathrm{MSE}_{\check{P}_b}(P_b)} \geq \frac{P_b}{\alpha(1-P_b)\log\frac{1}{1-P_b}} \tag{3.225}$$

$$\geq \frac{P_b}{(1-P_b)^{1-\alpha} - (1-P_b)}. \tag{3.226}$$

Clearly, the weaker bounds (3.224) and (3.226) are lower bounds for the corresponding limits of $\mathrm{MSE}_{\hat{P}_b}(P_b)/\mathrm{MSE}_{\check{\check{P}}_b}(P_b)$.

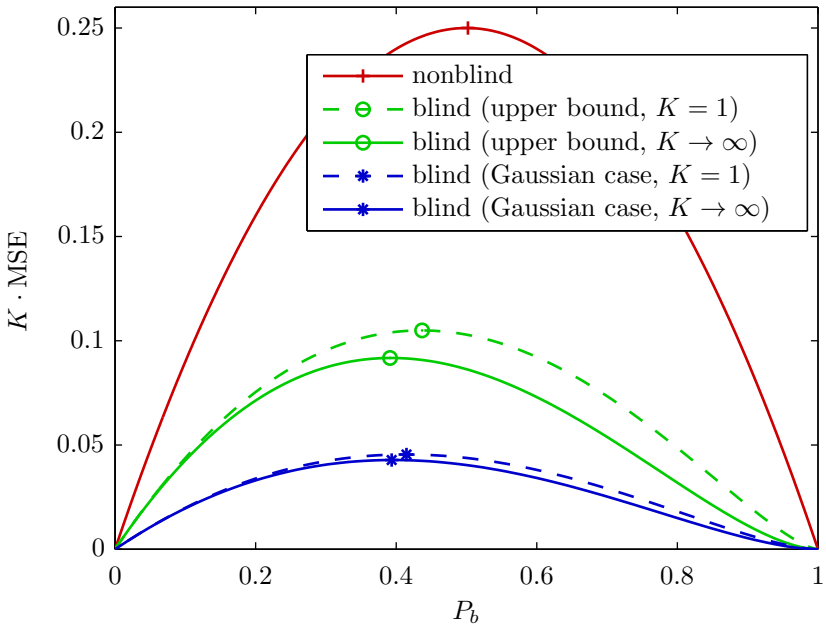


Figure 3.11: Block error probability ($N = 100$) in the binary case with MAP detection. Comparison of $\mathrm{MSE}_{\hat{P}_b}(P_b)$ (cf. (3.215)), upper bound (3.203), and $\mathrm{MSE}_{\check{P}_b}(P_b)$ (cf. (3.198)) in the Gaussian case.

As an example, we consider the binary case with MAP detection where $P_b = 1-(1-P_e)^N$ and $N = 100$. In Figure 3.11, we compare the MSE of the nonblind estimator $\check{P}_b$ to the MSE of the blind estimator $\hat{P}_b$ in the Gaussian case and to the upper bound (3.203) for $K = 1$ and $K \to \infty$. We observe that the blind estimator significantly outperforms the nonblind estimator for all P_b. Furthermore, $K\,\mathrm{MSE}_{\check{P}_b}(P_b)$ and the upper bound (3.203) show only a weak dependence on the sample size K. Figure 3.12 shows the MSE ratio $\mathrm{MSE}_{\hat{P}_b}(P_b)/\mathrm{MSE}_{\check{P}_b}(P_b)$ in the Gaussian case and compares it to the lower bound (3.216). The MSE ratio is at least equal to 2 which is in accordance to Proposition 3.11. In the Gaussian case, the blind estimator outperforms the corresponding nonblind estimator by a factor of more than 4 for small values of P_b. Finally, we note that the results for $\check{P}_b$ with $K = 1$ are equal to the results for $\check{\check{P}}_b$ with any $K \in \mathbb{N}$.

3.5.7 Minimum MSE

A blind estimator for the minimum MSE is given by

$$\check{\varepsilon}_{\mathrm{MMSE}} = \mathbb{E}\{\mathsf{u}^2\} - \frac{1}{K}\sum_{k=1}^{K}\left(\sum_{u\in\mathcal{U}} u\mathbb{P}\{\mathsf{u}=u|\mathbf{x}=\boldsymbol{x}_k\}\right)^2. \tag{3.227}$$

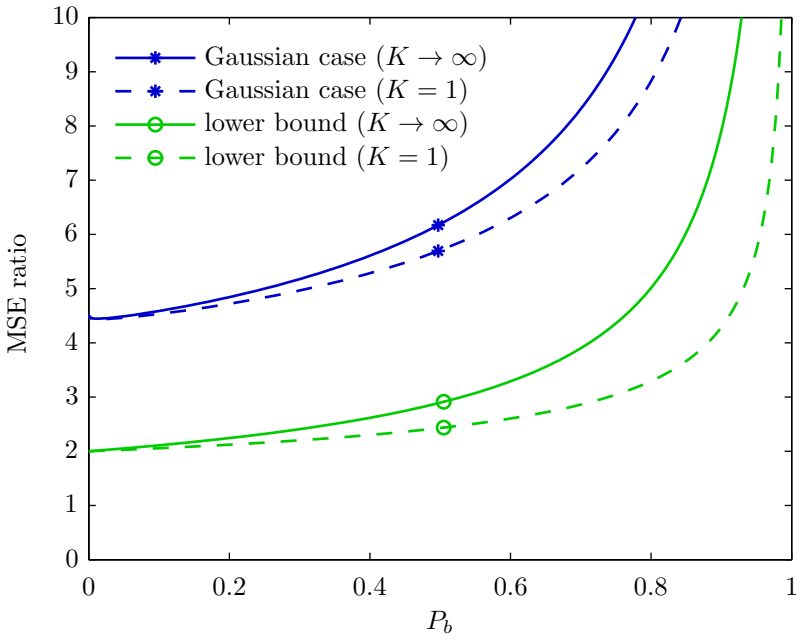


Figure 3.12: Block error probability ($N = 100$) in the binary case with MAP detection. Comparison of $\mathrm{MSE}_{\hat{P}_b}(P_b)/\mathrm{MSE}_{\check{P}_b}(P_b)$ in the Gaussian case to the lower bound (3.216).

This estimator is unbiased (cf. (3.105)) and its MSE equals

$$\mathrm{MSE}_{\check{\varepsilon}_{\mathrm{MMSE}}}(\varepsilon_{\mathrm{MMSE}}) = \mathbb{E}\{(\check{\varepsilon}_{\mathrm{MMSE}} - \varepsilon_{\mathrm{MMSE}})^2\} \tag{3.228}$$

$$= \frac{1}{K^2}\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{j=1}^{K}\left(\sum_{u\in\mathcal{U}} u\mathbb{P}\{\mathsf{u}=u|\mathbf{x}=\boldsymbol{x}_k\}\right)^2\left(\sum_{u\in\mathcal{U}} u\mathbb{P}\{\mathsf{u}=u|\mathbf{x}=\boldsymbol{x}_j\}\right)^2\right\}$$
$$- \left(\mathbb{E}\left\{\left(\sum\nolimits_{u\in\mathcal{U}} u\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\right)^2\right\}\right)^2 \tag{3.229}$$

$$= \frac{1}{K}\mathbb{E}\left\{\left(\sum\nolimits_{u\in\mathcal{U}} u\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\right)^4\right\} - \frac{1}{K}(\mathbb{E}\{\mathsf{u}^2\} - \varepsilon_{\mathrm{MMSE}})^2. \tag{3.230}$$

Using the inequality $\left(\sum_{u\in\mathcal{U}} u\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\right)^4 \leq \max_{u\in\mathcal{U}}|u|^2\left(\sum_{u\in\mathcal{U}} u\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\right)^2$, we upper bound the MSE as follows:

$$\mathrm{MSE}_{\check{\varepsilon}_{\mathrm{MMSE}}}(\varepsilon_{\mathrm{MMSE}}) \leq \frac{\max_{u\in\mathcal{U}}|u|^2}{K}\mathbb{E}\left\{\left(\sum\nolimits_{u\in\mathcal{U}} u\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\right)^2\right\} - \frac{1}{K}(\mathbb{E}\{\mathsf{u}^2\} - \varepsilon_{\mathrm{MMSE}})^2 \tag{3.231}$$

$$= \frac{\max_{u\in\mathcal{U}}|u|^2}{K}(\mathbb{E}\{\mathsf{u}^2\} - \varepsilon_{\mathrm{MMSE}}) - \frac{1}{K}(\mathbb{E}\{\mathsf{u}^2\} - \varepsilon_{\mathrm{MMSE}})^2. \tag{3.232}$$

In the binary case with $\mathsf{u} \in \{-1, 1\}$, we have

$$\mathrm{MSE}_{\check{\varepsilon}_{\mathrm{MMSE}}}(\varepsilon_{\mathrm{MMSE}}) = \mathbb{E}\{(\check{\varepsilon}_{\mathrm{MMSE}} - \varepsilon_{\mathrm{MMSE}})^2\} \tag{3.233}$$

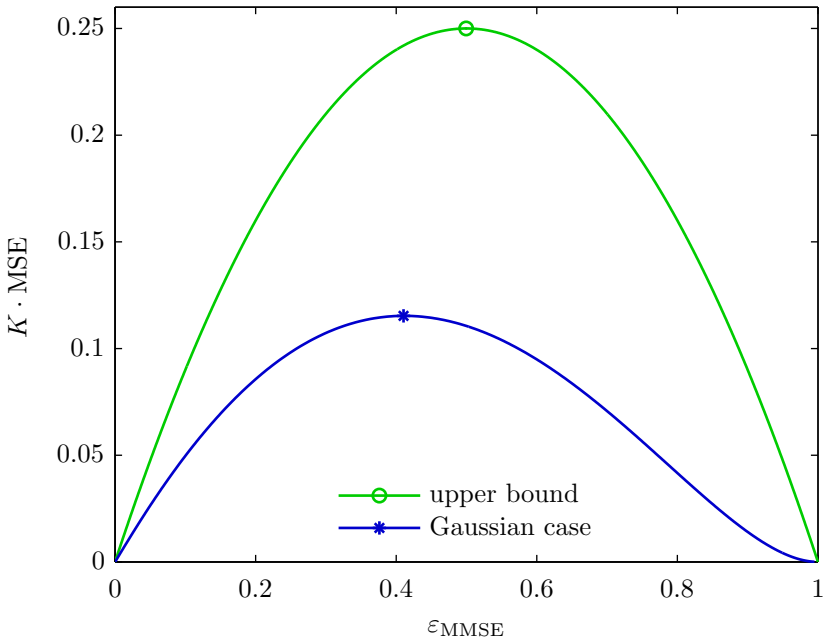


Figure 3.13: Minimum MSE estimation in the binary case. Comparison of the upper bound (3.237) to the MSE in the Gaussian case.

$$= \frac{1}{K^2} \sum_{k=1}^{K} \sum_{j=1}^{K} \mathbb{E}\{\Lambda_{\mathsf{u}}^2(\mathbf{x}_k)\Lambda_{\mathsf{u}}^2(\mathbf{x}_j)\} - \left(\mathbb{E}\{\Lambda_{\mathsf{u}}^2(\mathbf{x})\}\right)^2 \tag{3.234}$$

$$= \frac{1}{K}\mathbb{E}\{\Lambda_{\mathsf{u}}^4(\mathbf{x})\} - \frac{1}{K}\left(\mathbb{E}\{\Lambda_{\mathsf{u}}^2(\mathbf{x})\}\right)^2 \tag{3.235}$$

$$\leq \frac{1}{K}\mathbb{E}\{\Lambda_{\mathsf{u}}^2(\mathbf{x})\} - \frac{1}{K}\left(\mathbb{E}\{\Lambda_{\mathsf{u}}^2(\mathbf{x})\}\right)^2 \tag{3.236}$$

$$= \frac{\varepsilon_{\mathrm{MMSE}}}{K}(1 - \varepsilon_{\mathrm{MMSE}}). \tag{3.237}$$

The bound (3.236) is due to the fact that $|\Lambda_{\mathsf{u}}(\mathbf{x})| \leq 1$. In Figure 3.13, we plot the MSE (3.235) in the binary case with conditionally Gaussian LLRs and compare it to the upper bound (3.237).

3.5.8 Mutual Information and Conditional Entropy

The blind estimator

$$\check{H}(\mathsf{u}|\mathbf{x}) = -\frac{1}{K}\sum_{k=1}^{K}\sum_{u\in\mathcal{U}} \mathbb{P}\{\mathsf{u}=u|\mathbf{x}=\boldsymbol{x}_k\}\log_2 \mathbb{P}\{\mathsf{u}=u|\mathbf{x}=\boldsymbol{x}_k\}. \tag{3.238}$$

for the condition entropy $H(\mathsf{u}|\mathbf{x})$ is unbiased (cf. (3.115)) and we can write its MSE as follows:

$$\mathrm{MSE}_{\check{H}(\mathsf{u}|\mathbf{x})}\big(H(\mathsf{u}|\mathbf{x})\big) = \mathbb{E}\{\big(\check{H}(\mathsf{u}|\mathbf{x}) - H(\mathsf{u}|\mathbf{x})\big)^2\} \tag{3.239}$$

$$= \frac{1}{K}\mathbb{E}\Big\{\Big(\sum\nolimits_{u\in\mathcal{U}} \mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\log_2\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\Big)^2\Big\} - \frac{1}{K}H^2(\mathsf{u}|\mathbf{x}). \quad (3.240)$$

We use the inequality

$$\Big(\sum\nolimits_{u\in\mathcal{U}} \mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\log_2\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\Big)^2 \leq -H(\mathsf{u})\sum\nolimits_{u\in\mathcal{U}} \mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\log_2\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\} \quad (3.241)$$

to obtain the following upper bound for the MSE:

$$\mathrm{MSE}_{\check{H}(\mathsf{u}|\mathbf{x})}\big(H(\mathsf{u}|\mathbf{x})\big) \leq -\frac{H(\mathsf{u})}{K}\mathbb{E}\Big\{\sum\nolimits_{u\in\mathcal{U}} \mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\log_2\mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\Big\} - \frac{1}{K}H^2(\mathsf{u}|\mathbf{x}) \quad (3.242)$$

$$= \frac{H(\mathsf{u}|\mathbf{x})}{K}\big(H(\mathsf{u}) - H(\mathsf{u}|\mathbf{x})\big). \quad (3.243)$$

In the binary case with $\mathsf{u}\in\{-1,1\}$, we have

$$\mathrm{MSE}_{\check{H}(\mathsf{u}|\mathbf{x})}\big(H(\mathsf{u}|\mathbf{x})\big) = \mathbb{E}\{\big(\check{H}(\mathsf{u}|\mathbf{x}) - H(\mathsf{u}|\mathbf{x})\big)^2\} \quad (3.244)$$

$$= \frac{1}{K}\mathbb{E}\Big\{h_2^2\Big(\frac{1}{1+e^{|L_\mathsf{u}(\mathbf{x})|}}\Big)\Big\} - \frac{1}{K}H^2(\mathsf{u}|\mathbf{x}) \quad (3.245)$$

$$\leq \frac{H(\mathsf{u})}{K}\mathbb{E}\Big\{h_2\Big(\frac{1}{1+e^{|L_\mathsf{u}(\mathbf{x})|}}\Big)\Big\} - \frac{1}{K}H^2(\mathsf{u}|\mathbf{x}) \quad (3.246)$$

$$= \frac{H(\mathsf{u}|\mathbf{x})}{K}\big(H(\mathsf{u}) - H(\mathsf{u}|\mathbf{x})\big). \quad (3.247)$$

Replacing $H(\mathsf{u}|\mathbf{x})$ by $H(\mathsf{u}) - I(\mathsf{u};\mathbf{x})$ in the above expressions yields the MSE of the blind estimator for the mutual information $I(\mathsf{u};\mathbf{x})$. In Figure 3.14, we compare the MSE (3.245) in the binary case with conditionally Gaussian LLRs to the upper bound (3.247). Here, we assume a uniform prior, i.e., $H(\mathsf{u}) = 1$.

3.6 Cramér-Rao Lower Bound for Bit Error Probability Estimation

In this section we derive the CRLB for bit error probability estimation under MAP detection with conditionally Gaussian LLRs. Furthermore, we show that in this case there exists no efficient estimator. The importance of conditionally Gaussian LLRs has two main reasons: (i) the binary-input AWGN channel leads to conditionally Gaussian LLRs and (ii) numerous receiver algorithms use Gaussian approximations to reduce computational complexity. Therefore, we study the Gaussian case in more detail by analyzing the CRLB.

The distribution of the data L_u is given by

$$p(L_\mathsf{u};\mu) = p(L_\mathsf{u}|\mathsf{u}=1;\mu)\mathbb{P}\{\mathsf{u}=1\} + p(L_\mathsf{u}|\mathsf{u}=-1;\mu)\mathbb{P}\{\mathsf{u}=-1\} \quad (3.248)$$

$$= \frac{1}{\sqrt{4\pi\mu}}\left[\exp\Big(-\frac{1}{4\mu}(L_\mathsf{u}-\mu)^2\Big)\mathbb{P}\{\mathsf{u}=1\} + \exp\Big(-\frac{1}{4\mu}(L_\mathsf{u}+\mu)^2\Big)\mathbb{P}\{\mathsf{u}=-1\}\right], \quad (3.249)$$

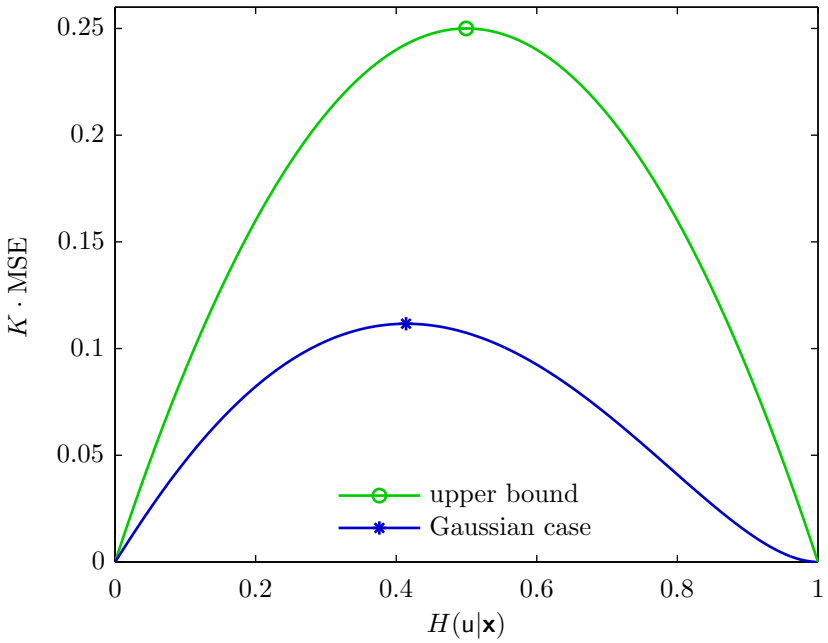


Figure 3.14: Estimation of $H(\mathsf{u}|\mathbf{x})$. Comparison of the MSE in the Gaussian case to the upper bound (3.247).

where $\mathsf{L}_\mathsf{u}|u \sim \mathcal{N}(u\mu, 2\mu)$ with $\mu > 0$.The parameter μ is related to the bit error probability P_e of the MAP detector as follows:

$$\mu(P_e) = 2Q^{-2}(P_e), \tag{3.250}$$

where $Q^{-m}(\cdot)$ is shorthand for $\big(Q^{-1}(\cdot)\big)^m$ and $Q^{-1}(\cdot)$ denotes the inverse of the Q-function. The following theorem states the CRLB in terms of P_e for the estimation problem defined by (3.249) and (3.250).

Theorem 3.12. *The CRLB for bit error probability estimation under MAP detection with K iid samples of conditionally Gaussian LLRs is given by*

$$MSE_{\check{P}_e}(P_e) = \mathrm{var}\{\check{P}_e\} \geq \frac{1}{K} \frac{Q^{-2}(P_e)}{4\pi \exp\big(Q^{-2}(P_e)\big)\big(1 + 2Q^{-2}(P_e)\big)}. \tag{3.251}$$

Proof: See Appendix A.5. ■

We note that the prior of u does not enter in the CRLB. In Figure 3.15, we plot the CRLB (3.251), the MSE of $\check{P}_e$ (cf. (3.185)), and the upper bound (3.186) for MAP detection. We observe that the MSE of $\check{P}_e$ does not attain the CRLB, i.e., the estimator $\check{P}_e$ is not efficient. Figure 3.16 depicts the comparison of Figure 3.15 with logarithmically scaled axes. We observe that the CRLB decays much more rapidly than the MSE of $\check{P}_e$ as $P_e \to 0$. The following theorem shows that an efficient estimator does not exist in the considered setting.

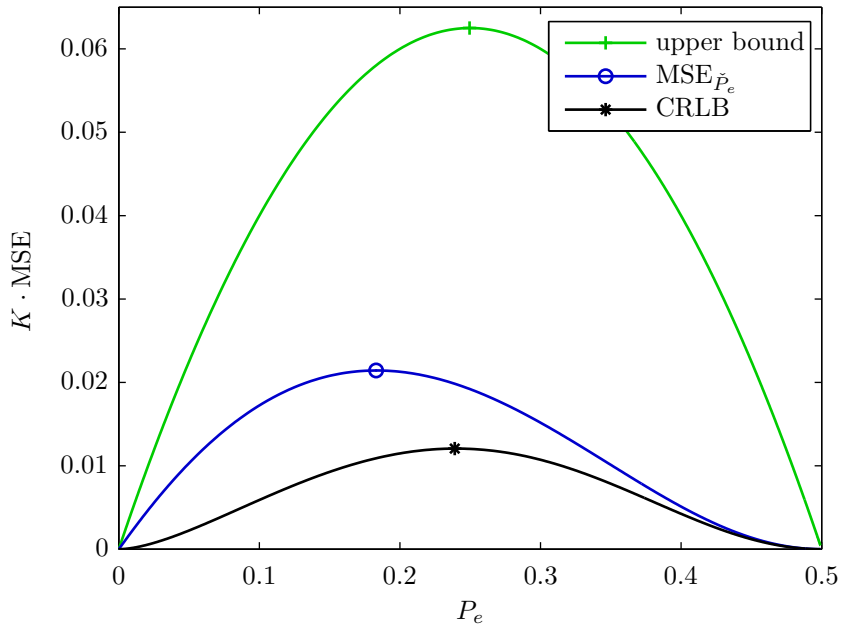


Figure 3.15: Comparison between CRLB (3.251), MSE of $\check{P}_e$ (cf. (3.185)) in the Gaussian case, and upper bound (3.186).

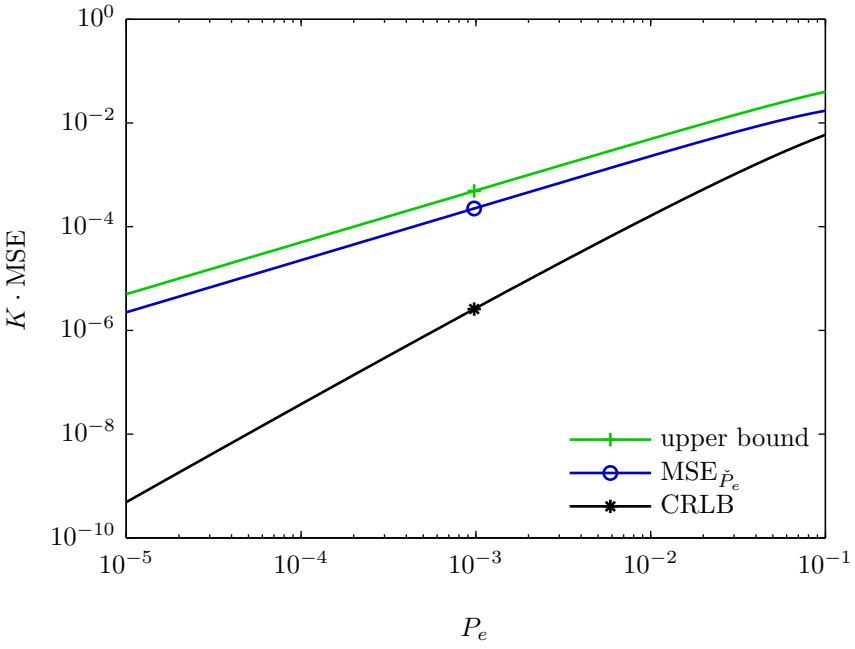


Figure 3.16: Comparison as in Figure 3.15 with logarithmically scaled axes.

Theorem 3.13. *For the problem of bit error probability estimation under MAP detection with conditionally Gaussian LLRs there exists no efficient estimator, i.e., the CRLB (3.251) cannot be attained uniformly.*

Proof: See Appendix A.6. ■

Although there exists no efficient estimator, we have shown in [107] that under certain conditions the estimator

$$\check{P}_e = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{1 + e^{|L_{\mathsf{u}}(\boldsymbol{x}_k)|}} \tag{3.252}$$

is the minimum-variance unbiased (MVU) estimator.

3.7 Application Examples and Approximate Log-Likelihood Ratios

We next give application examples in the communications context for some of the blind estimators proposed above. We include examples with suboptimal detection, approximate LLR computation, and model uncertainty to show that the proposed estimator are useful also in these cases.

3.7.1 MAP Detection

We first consider MAP detection of coded binary data which is transmitted over an AWGN channel. In particular, we assume a length-N block of data $\mathbf{u} = (\mathsf{u}_1 \ \ldots \ \mathsf{u}_N)^{\mathrm{T}}$ with $\mathbb{P}\{\mathsf{u}_n = 1\} = 1/2$, $n = 1, \ldots, N$. The data $\mathbf{u} \in \{0,1\}^N$ is channel-encoded, yielding a binary length-M codeword $\mathbf{c} = (\mathsf{c}_1 \ \ldots \ \mathsf{c}_M)^{\mathrm{T}}$. The codeword $\mathbf{c}$ is then BPSK modulated, i.e., the transmit signal equals $\mathbf{s} = \mathbf{1}_M - 2\mathbf{c}$. The output of the AWGN channel is thus given as

$$\mathbf{x} = \mathbf{s} + \mathbf{w}, \tag{3.253}$$

where $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ is iid circularly symmetric complex Gaussian noise with variance σ^2. Given the observed channel output $\boldsymbol{x}$, a MAP detector consists of the LLR computation

$$L_{\mathsf{c}_m}(x_m) = \frac{4}{\sigma^2} \Re(x_m), \quad m = 1, \ldots, M, \tag{3.254}$$

followed by soft-input channel decoding based on L_{c_m}, $m = 1, \ldots, M$. In the following, we assume that $N = 2^{10}$ and the channel code is a terminated $(7,5)_8$ convolutional code. In this setting, the BCJR algorithm (cf. Subsection 2.6.7) allows us to perform MAP-optimal soft-input detection. To obtain the results shown below, we have simulated $K = 10^6$ data blocks.

Figure 3.17 depicts the bit error rate (BER) versus SNR results obtained using the non-blind estimator $\hat{P}_e$ and the blind estimator $\check{P}_e$. Here, the BER represents the average of the N bit error probabilities $P_{e,n}$, $n = 1, \ldots, N$. We observe that both BER estimates are essen-

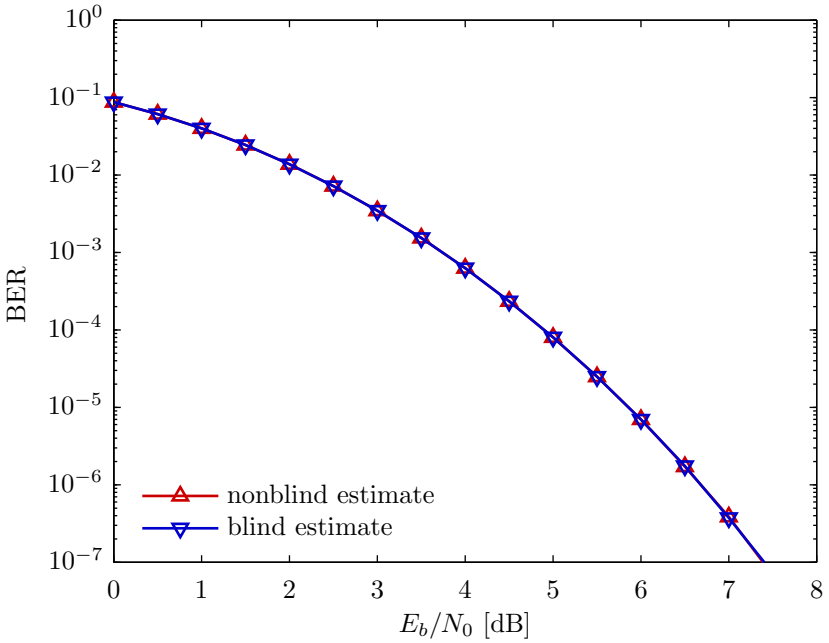


Figure 3.17: Comparison of nonblind and blind BER estimates under MAP detection.

tially equal which is in accordance with the results from Section 3.5. The blind estimator is unbiased since the posterior LLRs computed by the BCJR algorithm satisfy the consistency condition (3.25). Compared to the nonblind estimator, the blind estimator has a better MSE performance and thus converges faster. This can be used to either speed up simulations or to obtain more accurate results. Furthermore, the blind estimator is useful beyond computer simulations since the transmitted data is not required.

In Figure 3.18, we plot the frame error rate (FER), i.e., the block error probability for the entire block of data, versus the SNR. In this case, the blind estimators $\check{P}_b$ and $\check{\check{P}}_b$ are biased since they incorrectly assume independence of the individual bit errors. We observe that the blind estimators overestimate the FER. At an FER of 0.1, the blind estimates are approximately 0.2 dB away from the unbiased nonblind estimate and this gap vanishes as the SNR increases. In terms of the bias, it turns out that $\check{\check{P}}_b$ has a slightly smaller bias than $\check{P}_b$. We note that a cyclic redundancy check code can be used to approximately (i.e., ignoring undetected errors) implement $\hat{P}_b$ in a blind manner. However, the proposed blind estimators are attractive due to their faster convergence which may in practical applications outweigh the drawback of their small bias.

Figure 3.19 shows the upper bound $\check{\check{P}}_b^{+}$ as well as the lower bound $\check{\check{P}}_b^{-}$ (cf. (3.102)) and compares them to $\check{\check{P}}_b$. At an FER of 0.1, the bounds are approximately 0.4 dB apart and they converge as the SNR increases. This confirms the usefulness of these bounds. We note that the upper bound is closer to the blind estimate than the lower bound. Furthermore, the lower bound $\check{\check{P}}_b^{-}$ is significantly tighter than $\check{P}_b^{-}$ (cf. (3.99)).

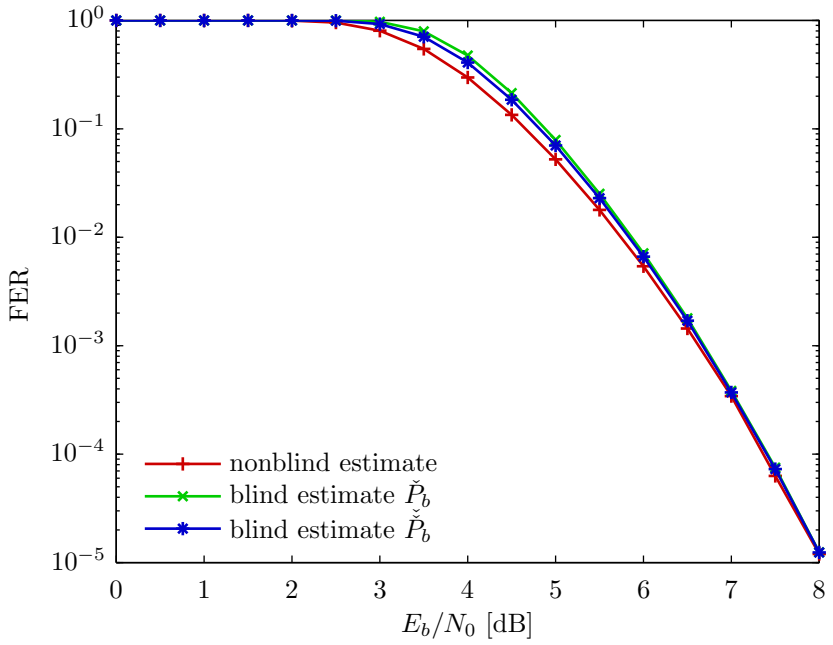


Figure 3.18: Comparison of nonblind and blind FER estimates under MAP detection. In contrast to the blind estimates, the nonblind estimate is unbiased.

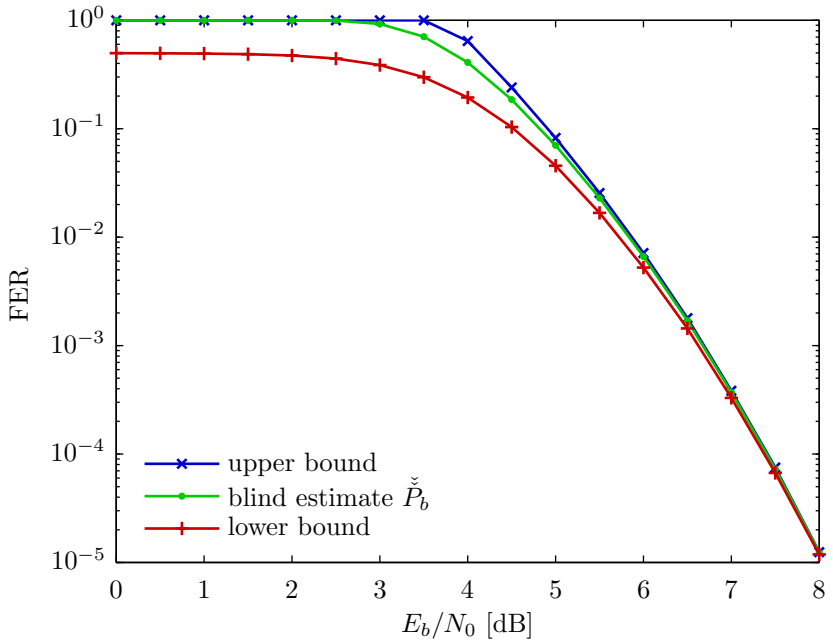


Figure 3.19: Comparison of the blind estimate $\check{\check{P}}_b$ and the bounds (3.102).

3.7.2 Approximate MAP Detection

We next consider data transmission over a fading channel using bit-interleaved coded modulation (BICM). As above, the binary data $\mathbf{u} \in \{0,1\}^N$ is channel-encoded yielding a length-M codeword $\mathbf{c}$. The interleaved codeword $\mathbf{c}' = \Pi(\mathbf{c})$ is then mapped onto a 16-QAM signal constellation $\mathcal{A}$ with Gray labeling. We denote the transmit signal of length $J = \lceil M/4 \rceil$ by $\mathbf{s} = \varphi(\mathbf{c}') \in \mathcal{A}^J$. The output of the fading channel is given as (here, $\odot$ denotes element-wise multiplication)

$$\mathbf{x} = \boldsymbol{h} \odot \mathbf{s} + \mathbf{w}, \tag{3.255}$$

where $\boldsymbol{h}$ denotes the vector of channel coefficients and $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ is iid circularly symmetric complex Gaussian noise with variance σ^2. We assume that the receiver has perfect channel state information (CSI), i.e., $\boldsymbol{h}$ is known. Given the observed channel output $\boldsymbol{x}$, the BICM receiver first computes the LLRs

$$L_{\mathsf{c}'_{4(j-1)+i}}(x_j) = \log \frac{\sum_{s \in \mathcal{A}_i^0} \exp\bigl(-|x_j - h_j s|^2/\sigma^2\bigr)}{\sum_{s \in \mathcal{A}_i^1} \exp(-|x_j - h_j s|^2/\sigma^2)}, \qquad j = 1, \ldots, J, \quad i = 1, \ldots, 4, \tag{3.256}$$

where $\mathcal{A}_i^b$ denotes the set of symbols whose bit label at position i is equal to b (note that $\mathcal{A}_i^0 \cup \mathcal{A}_i^1 = \mathcal{A}$ and $\mathcal{A}_i^0 \cap \mathcal{A}_i^1 = \emptyset$). The deinterleaved LLRs $L_{\mathsf{c}_m} = \Pi^{-1}(L_{\mathsf{c}'_m})$, $m = 1, \ldots, M$, are then used to decode the data. The BICM receiver is inherently suboptimal because the channel decoder incorrectly treats the code bits that are mapped to the same symbol as if they were conditionally independent. In the following, we assume that $N = 2^{12}$ and the channel code is a terminated $(37, 21)_8$ convolutional code. The realizations of the channel coefficients are drawn from a circularly symmetric complex Gaussian distribution with unit variance. The channel remains constant for 10 symbols and the different channel coefficients are independent. To obtain the results shown below, we have simulated $K = 10^5$ data blocks.

Figure 3.20 shows the BER versus SNR results obtained using the nonblind estimator $\hat{P}_e$ and the blind estimator $\check{P}_e$. Although the BICM receiver is suboptimal, we observe that both BER estimates are essentially equal. The simplifying independence assumption of the BICM receiver entails a performance penalty since statistical dependencies are ignored. However, the LLRs computed by the receiver satisfy the consistency condition (3.25) and thus the blind BER estimate is unbiased.

A relevant code-independent performance measure for BICM systems is [27]

$$C_{\mathrm{BICM}} = \sum_{i=1}^{|\mathcal{A}|} I(\mathsf{c}_i; \mathsf{L}_{\mathsf{c}_i}), \tag{3.257}$$

which can be viewed as the capacity of an equivalent modulation channel with inputs c_i and outputs L_{c_i}, $i = 1, \ldots, |\mathcal{A}|$. In Figure 3.21, we compare a nonblind estimate of C_{BICM} (obtained using histograms of the conditional LLR distributions) to a blind estimate (obtained using the blind estimator $\check{I}(\mathsf{c}_i; \mathsf{L}_{\mathsf{c}_i})$). We note that both estimates are essentially equal which

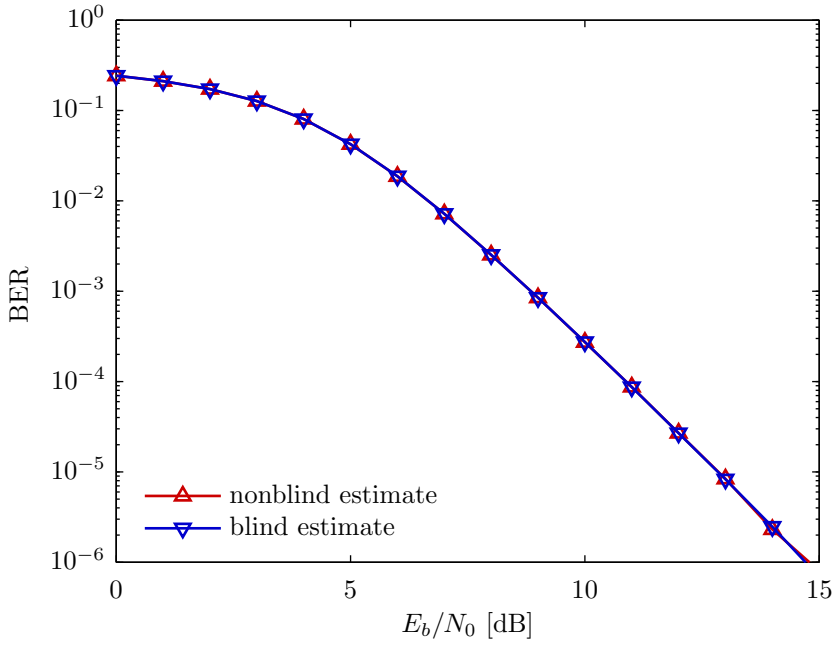


Figure 3.20: Comparison of nonblind and blind BER estimates for BICM transmission over a block fading channel.

is again due to the consistency of the LLRs. The blind estimator is attractive in this case not only due to its faster convergence but also since it avoids the estimation of the conditional LLR densities.

The independence assumption of BICM receivers makes them inherently mismatched. Another source of mismatch in practical BICM systems is due to approximate LLR computation in the demodulator and the channel decoder. Due to these approximations the consistency condition is violated and the blind estimates become biased. However, this source of mismatch can be eliminated by performing LLR correction [47] which makes the blind estimates unbiased. Unfortunately, LLR correction cannot be performed in a blind manner, i.e., some side information is necessary for LLR correction.

3.7.3 Iterative Detection

In this subsection, we consider channel-coded data transmission with iterative decoding (cf. Subsection 2.6.5) at the receiver. The difference between iterative channel decoding and the above BICM example is that an iterative decoder always computes approximate LLRs. This is due to the feedback of extrinsic LLRs which are in each iteration incorrectly assumed to be independent. Therefore, we cannot expect the blind estimators to be unbiased. In the examples below, we consider an AWGN channel with BPSK-modulated input as in Subsection 3.7.1. Using the channel output $\boldsymbol{x}$, the LLRs $L_{\mathsf{c}_m}(x_m)$, $m = 1, \ldots, M$, are computed according to (3.254). We have simulated $K = 10^5$ data blocks to obtain the results shown below.

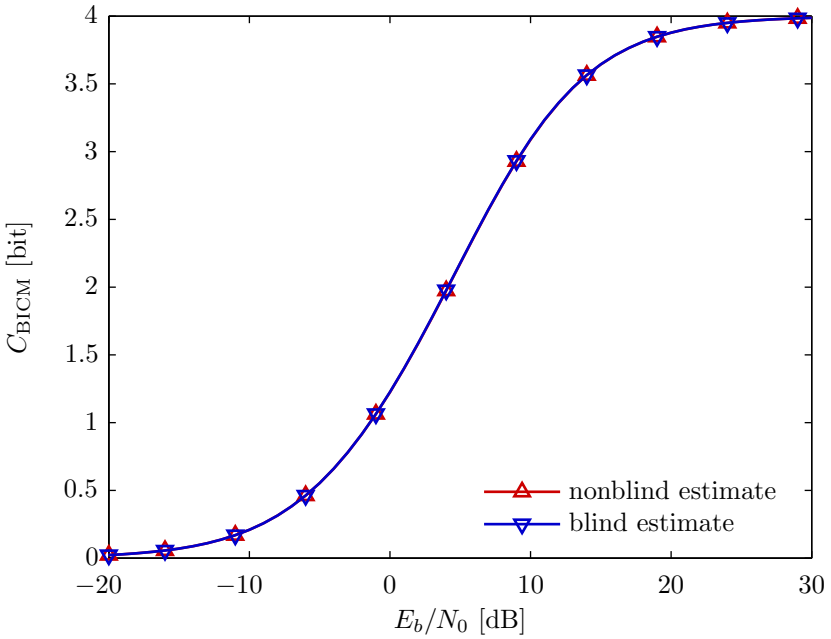


Figure 3.21: Comparison of nonblind and blind estimates of C_{BICM}.

Figure 3.22 shows the BER performance of a rate-1/2 irregular low-density parity-check (LDPC) code[4] with a blocklength of 64000 bits. The solid lines depict the performance estimates for a belief propagation (BP) decoder and the dashed lines correspond to the BER estimates for a min-sum decoder. Both decoders execute up to 100 iterations with a parity check after each iteration. We observe that the blind estimate for the BER of the BP decoder matches the unbiased nonblind estimate very well, except in the error floor regime. The blind BER estimate for the min-sum decoder overestimates the performance for SNRs below the threshold and it matches the nonblind estimate in the waterfall regime.

In Figure 3.23, we depict the BER versus SNR results for a rate-1/2 turbo code with $N = 2^{16}$ information bits and 10 decoder iterations. Here, the iterative decoder uses a max-log-MAP decoder to decode the constituent $(37, 21)_8$ codes. We observe that the blind estimate again matches the unbiased nonblind estimate very well in the waterfall regime. In the error floor regime, the blind estimate again suffers from a significant bias.

An important issue with iterative decoders is LLR clipping. The effect of LLR clipping on the blind estimates is quite severe, e.g., if the maximum magnitude of all LLRs is 10, then the minimum blind BER estimate equals approximately $4.5 \cdot 10^{-5}$, although the true BER may be much smaller. Clearly, LLR clipping entails an underestimation of the reliability which can be accounted for by LLR correction [91]. From the above results we can conclude that the blind estimator $\check{P}_e$ is useful for iterative decoders in the waterfall regime, i.e., for

[4]We note that this particular code is optimized for a small decoding threshold which results in a rather pronounced error floor.

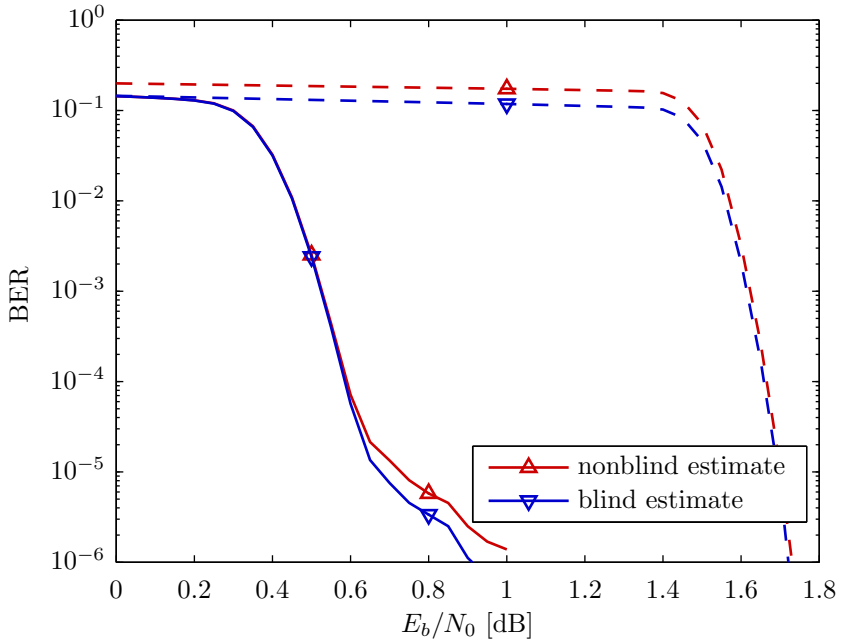


Figure 3.22: Comparison of nonblind and blind BER estimates for BP decoding (solid lines) and min-sum decoding (dashed lines) of an irregular LDPC code. The nonblind estimate is unbiased.

BER values from, say, 10^{-1} down to 10^{-5}. This corresponds to the BER values of interest in many applications. We note that the proposed blind estimator is not suitable for estimating the error floor performance of iterative decoders.

3.7.4 Imperfect Channel State Information

Finally, we consider the case of data detection with imperfect CSI due to channel estimation errors. We use BICM with a terminated $(37, 21)_8$ convolutional code and a BPSK signal constellation. The binary data $\mathbf{u} \in \{0,1\}^N$, with $N = 2^{12}$, is channel-encoded yielding a codeword $\mathbf{c}$ of length $M = 8200$. The interleaved codeword $\mathbf{c}' = \Pi(\mathbf{c})$ is then BPSK-modulated, i.e., the transmit signal is given as $\mathbf{s} = \mathbf{1}_M - 2\mathbf{c}'$. The input-output relation of the channel is given as

$$\mathbf{x} = \mathbf{h} \odot \mathbf{s} + \mathbf{w}, \tag{3.258}$$

where $\mathbf{h}$ is the vector of channel coefficients and $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ is iid circularly symmetric complex Gaussian noise with variance σ^2. We assume that the channel remains constant for 10 symbols and the different channel coefficients are independent. Therefore, we can write the vector of channel coefficients as

$$\mathbf{h} = \mathrm{vec}\left\{\mathbf{1}_{10}\tilde{\mathbf{h}}^{\mathrm{T}}\right\}, \tag{3.259}$$

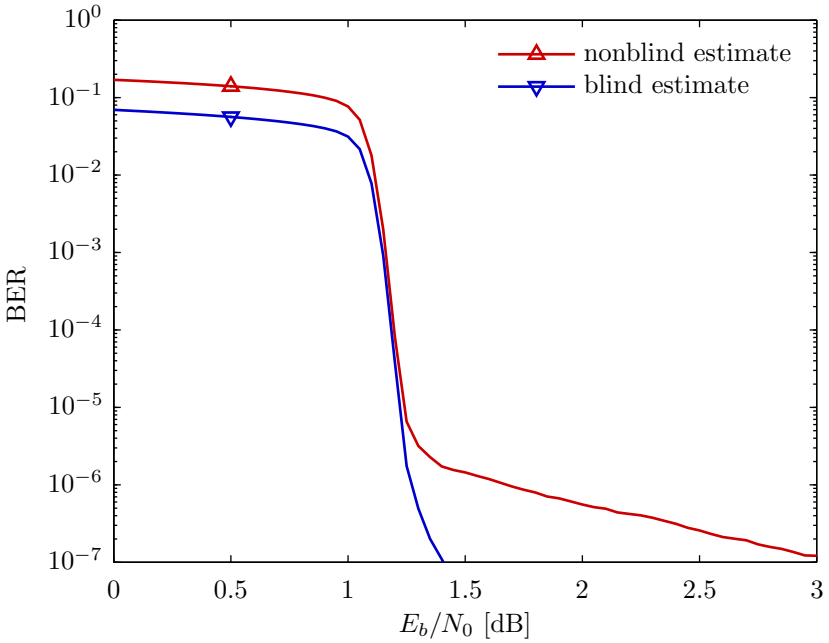


Figure 3.23: Comparison of nonblind and blind BER estimates for max-log-MAP decoding of a turbo code. The nonblind estimate is unbiased.

where $\widetilde{\mathbf{h}} \sim \mathcal{CN}(\mathbf{0}, \boldsymbol{I})$ are the independently drawn channel coefficients and the operator vec$\{\cdot\}$ concatenates the columns of its argument matrix. The first symbol of each length-10 block is a pilot symbol. In total we estimate the $M/10$ channel coefficients $\widetilde{\mathsf{h}}_k$ using the pilot symbols s_k, $k = 1, \ldots, M/10$. The minimum MSE estimate of $\widetilde{\mathsf{h}}_k$ equals

$$\hat{\widetilde{h}}_k(x_{10(k-1)+1}) = \frac{s_k x_{10(k-1)+1}}{1+\sigma^2}, \quad k = 1, \ldots, M/10. \tag{3.260}$$

Using (3.260), we can write the vector of estimated channel coefficients as follows:

$$\hat{\boldsymbol{h}} = \mathrm{vec}\left\{ \mathbf{1}_{10} \hat{\widetilde{\boldsymbol{h}}}^{\mathrm{T}} \right\}. \tag{3.261}$$

A mismatched detector uses the estimated channel coefficients as if they were the true channel coefficients. Hence, a mismatched detector computes the LLRs

$$L^{\mathrm{mis}}_{\mathsf{c}'_m}(x_m) = \frac{4}{\sigma^2} \Re\left(x_m \hat{h}^*_m \right), \tag{3.262}$$

where $\hat{h}^*_m$ denotes the complex conjugate of $\hat{h}_m$. A matched detector computes the following LLRs [79]:

$$L_{\mathsf{c}'_m}(x_m) = \log \frac{p(x_m | \hat{h}_m, s_m = 1)}{p(x_m | \hat{h}_m, s_m = -1)}, \tag{3.263}$$

where

$$p(x_m|\hat{h}_m, s_m) = \int_{-\infty}^{\infty} p(x_m|h_m, s_m)p(h_m|\hat{h}_m)\mathrm{d}h_m. \tag{3.264}$$

Using the Schur complement we find that $\mathsf{h}_m|\hat{h}_m \sim \mathcal{CN}\Big(\hat{h}_m, \frac{\sigma^2}{1+\sigma^2}\Big)$. Furthermore, we have $\mathsf{x}_m|\hat{h}_m, s_m \sim \mathcal{CN}(\hat{h}_m s_m, \sigma^2)$. Evaluating the integral in (3.264) yields

$$\mathsf{x}_m|\hat{h}_m, s_m \sim \mathcal{CN}\left(\hat{h}_m s_m, \sigma^2 + \frac{\sigma^2}{1+\sigma^2}\right). \tag{3.265}$$

Using (3.265) in (3.263), we obtain

$$L_{\mathsf{c}'_m}(x_m) = \frac{1+\sigma^2}{2+\sigma^2}\frac{4}{\sigma^2}\Re\Big(x_m\hat{h}_m^*\Big) = \frac{1+\sigma^2}{2+\sigma^2}L_{\mathsf{c}'_m}^{\mathrm{mis}}(x_m). \tag{3.266}$$

Hence, the LLRs computed by the matched detector are a downscaled version of the mismatched detector's LLRs. In what follows, we compare the BER estimates in the matched and the mismatched case. To obtain the results shown below, we have simulated $K = 10^5$ data blocks.

The red curve ('△' markers) in Figure 3.24 corresponds to the unbiased nonblind BER estimate for the matched detector. We do not plot the BER performance of the mismatched detector since it is virtually indistinguishable from the performance of the matched detector. The blind BER estimate for the mismatched detector (dashed blue line with '▽' markers in Figure 3.24) is biased since the consistency condition is violated if $\hat{h}_m \neq h_m$ in (3.262). Using the matched detector with the LLRs (3.266) effectively eliminates the bias of the blind BER estimate (cf. solid blue line with '▽' markers in Figure 3.24).

We note that for a BPSK signal constellation the BICM receiver is MAP-optimal, i.e., the only source of mismatch is due to channel estimation errors. Furthermore, the LLR computation (3.266) of the matched detector cannot be viewed as an LLR correction applied to $L_{\mathsf{c}'_m}^{\mathrm{mis}}(x_m)$. In fact, the LLRs in (3.266) do not satisfy the consistency condition in general (this is the reason for the very minor bias of the corresponding blind BER estimate). Moreover, we cannot perform LLR correction in this case since that would require the true channel coefficients which are not available. However, the results in Figure 3.24 show that blind estimation can yield accurate results in the presence of channel estimation errors.

3.8 Discussion

In this chapter, we have proposed blind estimators for the (conditional) error probabilities, the minimum MSE, and the mutual information in the context of Bayesian hypothesis tests. We have analyzed and suitably bounded the MSE of the blind estimators and we have included a comparison to nonblind estimators for the (conditional) error probabilities. For the unconditional error probability we have proven that the blind estimator always dominates the nonblind estimator. Furthermore, for the conditional error probabilities, we have given

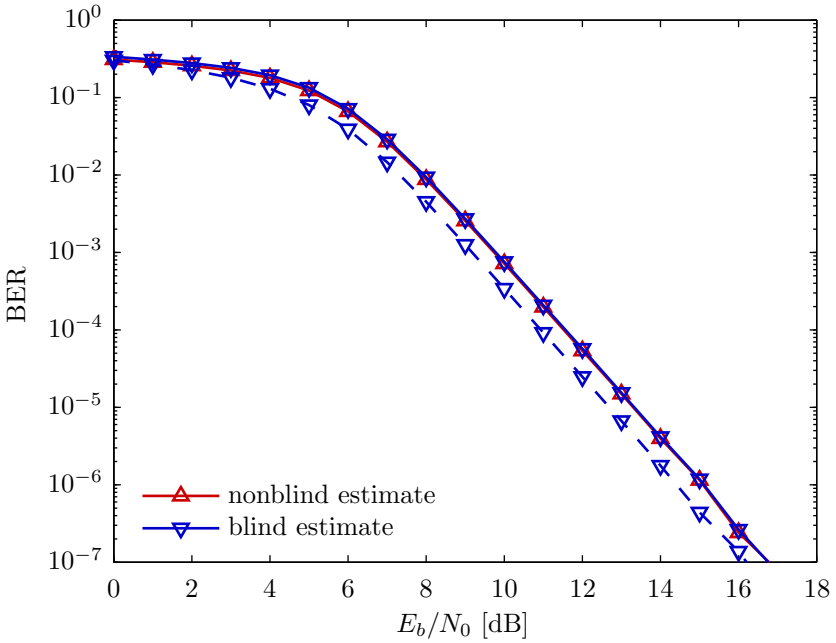


Figure 3.24: Comparison of nonblind and blind BER estimates in the presence of channel estimation errors. The solid lines correspond to the results for the matched detector and the dashed lines is obtained using the mismatched detector.

conditions under which the blind estimators dominate the corresponding nonblind estimators for all distributions of the data. However, when we consider specific and relevant distributions of the data (e.g., the Gaussian case), all blind estimators significantly outperform the corresponding nonblind estimators for relevant parameter values. For the case of bit error probability estimation with conditionally Gaussian LLRs, the MSE of the blind estimator is more than 4 times smaller than the MSE of the nonblind estimator. The blind estimators can thus be used to obtain more accurate results or to speed-up computer simulations while maintaining the required accuracy.

In the case of binary hypothesis tests, our results are based on the LLR properties we have studied in Section 3.3. We have shown that the consistency condition connects the conditional pdfs $p(L_{\mathsf{u}}|\mathsf{u}=u)$, $u \in \{-1, 1\}$, and the unconditional pdf $p(L_{\mathsf{u}})$ such that any one of the three is sufficient to determine the other two. We shall show that this property of LLR distributions is important also in the context of quantizer design.

We note that further relevant performance metrics can be estimated in a blind manner. For example, the Bayesian risk associated to the detector $\hat{u}$ can be written as

$$R(\hat{u}) = \sum_{u \in \mathcal{U}} \mathbb{E}\big\{C_{\hat{u}(\mathbf{x}),u} \mathbb{P}\{\mathsf{u}=u|\mathbf{x}\}\big\}, \tag{3.267}$$

where $C_{u',u} \geq 0$ is the cost of the decision $\hat{u} = u'$ when $\mathsf{u} = u$. Similarly, in the binary

case ($\mathsf{u} \in \{-1, 1\}$) with uniform prior probabilities we can write the relative entropy between $p(L_\mathsf{u}|\mathsf{u}=u)$ and $p(L_\mathsf{u}|\mathsf{u}=-u)$ as follows:

$$D\big(p(L_\mathsf{u}|\mathsf{u}=u)\|p(L_\mathsf{u}|\mathsf{u}=-u)\big) = 2\mathbb{E}\{\mathsf{L}_\mathsf{u}\Lambda_\mathsf{u}\}. \tag{3.268}$$

We note that the right-hand side of (3.268) does not depend on u. Furthermore, in this case the deflection [99] with the soft bit Λ_u as test statistic equals

$$d^2_{\Lambda_\mathsf{u}} = \frac{4\mathbb{E}\{\Lambda^2_\mathsf{u}\}}{1-\mathbb{E}\{\Lambda^2_\mathsf{u}\}}. \tag{3.269}$$

Replacing the expectations in (3.267)-(3.269) by sample means yields blind estimators for the respective quantities. We thereby obtain unbiased blind estimators for the Bayesian risk and the relative entropy, and an asymptotically unbiased blind estimator for the deflection.

We have derived the CRLB for bit error probability estimation with conditionally Gaussian LLRs under MAP detection. We find that the proposed blind estimator for the bit error probability is not efficient. Moreover, we were able to prove that an efficient estimator does not exist for this estimation problem. Comparing the CRLB to the MSE shows that the gap to the CRLB increases as the error probability goes to zero. However, we have shown in [107] that in certain case our blind estimator is the MVU estimator.

Finally, the numerical results in Section 3.7 confirm that the blind estimators proposed in this chapter are useful also when suboptimal detectors are used and when the data model is not exact. This is especially important in the communications setting, where channel estimation errors are unavoidable in practice. We conclude that our blind estimators are suitable for online performance estimation of Bayesian detectors without training data overhead.

4

The Rate-Information Trade-off in the Gaussian Case

In this chapter, we discuss the trade-off between quantization rate and relevant information for jointly Gaussian random variables. We introduce the problem setting and provide the required background in Section 4.1. In Section 4.2, we formalize the rate-information trade-off and we define the information-rate function and the rate-information function. A review of the Gaussian information bottleneck (GIB) is given in Section 4.3. We then use the GIB to derive closed-form expressions for the rate-information trade-off in the univariate case (Section 4.4) and in the multivariate case (Section 4.5). Next, we study the connection between the rate-information trade-off and the rate-distortion (RD) trade-off in Section 4.6. We show that optimal quantization with respect to the mean-square error (MSE) is rate-information-optimal if suitable linear preprocessing is performed. In Section 4.7, we design quantizers and we compare their performance to the optimal rate-information trade-off. We conclude this chapter with a discussion of our results and we mention possible extensions in Section 4.8.

4.1 Introduction and Background

We consider jointly Gaussian random vectors x and y and we are interested in compressing y such that its compressed version z contains as much information about x as possible. More specifically, we want to find the optimal trade-off between the compression rate $I(\mathsf{y};\mathsf{z})$ and the relevant information $I(\mathsf{x};\mathsf{z})$. We term this trade-off the *rate-information trade-off* (cf. Section 4.2). Note that in this context, we use the terms *quantization* and *compression* interchangeably.

Our motivation for studying this trade-off stems from communication theory. In a communications setting, it is of great interest to find the largest data rate at which we can reliably transmit over a channel whose output is quantized with a certain compression rate. However, we emphasize that our results apply to arbitrary jointly Gaussian data sets and are not restricted to the communication setting.

Finding the rate-information trade-off basically amounts to solving the information bottleneck (IB) problem (2.128). Unfortunately, in the Gaussian case we cannot determine the rate-information trade-off using the iterative IB algorithm (cf. Algorithm 2.2) since it is restricted to discrete random variables. However, the GIB (cf. Section 4.3) allows us to find closed-form expressions for the rate-information trade-off in the Gaussian case. In the following we assume that $\mathsf{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_\mathsf{x})$ and $\mathsf{y} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_\mathsf{y})$ are zero-mean and $\boldsymbol{C}_\mathsf{y}$ has full rank[1]. We note that any zero-mean and jointly Gaussian x, y can be written as [8, Theorem 4.5.5]

$$\mathsf{y} = \boldsymbol{H}\mathsf{x} + \mathsf{w}, \tag{4.1}$$

with a deterministic matrix $\boldsymbol{H}$ and a Gaussian random vector $\mathsf{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_\mathsf{w})$ which is independent of x. In the following we work with the linear model (4.1) which can also be viewed as the input-output relation of a constant multiple-input multiple-output channel.

Next, we recall the definition of the rate-distortion function and the distortion-rate function [20, Section 10.2].

Definition 4.1. *The rate-distortion function for a source* y *with distortion measure* $d(y, \hat{y})$ *is defined as*

$$R(D) \triangleq \min_{p(\hat{y}|y)} I(\mathsf{y};\hat{\mathsf{y}}) \qquad \textit{subject to} \quad \mathbb{E}\{d(\mathsf{y},\hat{\mathsf{y}})\} \leq D, \tag{4.2}$$

and the distortion-rate function is defined as

$$D(R) \triangleq \min_{p(\hat{y}|y)} \mathbb{E}\{d(\mathsf{y},\hat{\mathsf{y}})\} \qquad \textit{subject to} \quad I(\mathsf{y};\hat{\mathsf{y}}) \leq R. \tag{4.3}$$

The rate-distortion function $R(D)$ quantifies the minimum rate required to reconstruct the source y with an average distortion not exceeding D. Similarly, $D(R)$ quantifies the minimum average distortion incurred for compressing the source y with a rate of at most R.

[1] These assumptions are not restrictive, since otherwise x and y can be centered and y can be reduced to the $\operatorname{rank}(\boldsymbol{C}_\mathsf{y})$-dimensional subspace on which its distribution is supported.

For most RD problems of interest, closed-form expressions for the rate-distortion function or the distortion-rate function are not available. However, for a Gaussian source $\mathsf{y} \sim \mathcal{N}(0, \sigma^2)$ and squared-error distortion $d(y, \hat{y}) = (y - \hat{y})^2$ we have [20, Section 10.3]

$$R(D) = \frac{1}{2} \log_2^+ \frac{\sigma^2}{D}, \tag{4.4}$$

and

$$D(R) = 2^{-2R} \sigma^2. \tag{4.5}$$

We note that (4.4) and (4.5) establish the fundamental performance limits for MSE-optimal source coding of Gaussian sources.

In a *noisy* source coding problem, a noisy version $\tilde{\mathsf{y}}$ of the source signal y is quantized. In this case, when y and $\tilde{\mathsf{y}}$ are jointly Gaussian, the MSE-optimal strategy is to estimate y from $\tilde{\mathsf{y}}$ using a Wiener filter whose output $\check{\mathsf{y}}$ (which is again Gaussian) is then quantized in an MSE-optimal manner. The corresponding overall MSE is the sum of the MSEs due to the estimation and the quantization, respectively [8, Subsection 4.5.4].

We next explain MSE-optimal noisy source coding in terms of a simple example. Let the source $\mathsf{y} \sim \mathcal{N}(0, \sigma^2)$ be transmitted over a Gaussian channel such that $\tilde{\mathsf{y}} = \mathsf{y} + \mathsf{w}$, where $\mathsf{w} \sim \mathcal{N}(0, \sigma_{\mathsf{w}}^2)$ is independent of y. Then we have $\tilde{\mathsf{y}} \sim \mathcal{N}(0, \sigma^2 + \sigma_{\mathsf{w}}^2)$ and the minimum MSE estimate of y given $\tilde{\mathsf{y}}$ is given as

$$\check{\mathsf{y}} = \frac{\sigma^2}{\sigma^2 + \sigma_{\mathsf{w}}^2} \tilde{\mathsf{y}}, \tag{4.6}$$

where $\check{\mathsf{y}}$ is Gaussian with variance $\sigma^2/(1 + \sigma_{\mathsf{w}}^2/\sigma^2)$. The MSE of the estimator in (4.6) equals

$$\mathbb{E}\{(\mathsf{y} - \check{\mathsf{y}})^2\} = \sigma^2 - \frac{\sigma^2}{1 + \sigma_{\mathsf{w}}^2/\sigma^2}. \tag{4.7}$$

MSE-optimal quantization of $\check{\mathsf{y}}$ with rate R yields an MSE distortion of (cf. (4.5))

$$\mathbb{E}\{(\check{\mathsf{y}} - \hat{\mathsf{y}})^2\} = 2^{-2R} \frac{\sigma^2}{1 + \sigma_{\mathsf{w}}^2/\sigma^2}. \tag{4.8}$$

The overall MSE is given by

$$\begin{aligned} \mathbb{E}\{(\mathsf{y} - \hat{\mathsf{y}})^2\} &= \mathbb{E}\{(\mathsf{y} - \check{\mathsf{y}} + \check{\mathsf{y}} - \hat{\mathsf{y}})^2\} && (4.9) \\ &= \mathbb{E}\{(\mathsf{y} - \check{\mathsf{y}})^2\} + \mathbb{E}\{(\check{\mathsf{y}} - \hat{\mathsf{y}})^2\} && (4.10) \\ &= \sigma^2 - (1 - 2^{-2R}) \frac{\sigma^2}{1 + \sigma_{\mathsf{w}}^2/\sigma^2}. && (4.11) \end{aligned}$$

Here, (4.10) is due to the orthogonality principle, i.e., the estimation error $\mathsf{y} - \check{\mathsf{y}}$ is orthogonal to any function of the observation $\tilde{\mathsf{y}}$. Hence, the overall MSE (4.11) is indeed the sum of the MSEs in (4.7) and (4.8).

Finally, we note that the rate-distortion function of a Gaussian vector source $\mathbf{y}$ with squared-error distortion $d(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ can be written as a sum of rate-distortion functions as in (4.4). Here, the appropriate distortion values are found by performing reverse waterfilling on the eigenvalues of the covariance matrix of $\mathbf{y}$ [20, Section 10.3].

4.2 The Rate-Information Trade-off

We next formalize the trade-off between compression rate and relevant information. To this end, we define the information-rate function $I(R)$ and the rate-information function $R(I)$.

Definition 4.2. *Let $\mathbf{x} \leftrightarrow \mathbf{y} \leftrightarrow \mathbf{z}$ be a Markov chain. The information-rate function $I\colon \mathbb{R}_+ \to [0, I(\mathbf{x};\mathbf{y})]$ is defined as*

$$I(R) \triangleq \max_{p(\boldsymbol{z}|\boldsymbol{y})} I(\mathbf{x};\mathbf{z}) \qquad \text{subject to} \quad I(\mathbf{y};\mathbf{z}) \leq R, \tag{4.12}$$

and the rate-information function $R\colon [0, I(\mathbf{x};\mathbf{y})] \to \mathbb{R}_+$ is defined as

$$R(I) \triangleq \min_{p(\boldsymbol{z}|\boldsymbol{y})} I(\mathbf{y};\mathbf{z}) \qquad \text{subject to} \quad I(\mathbf{x};\mathbf{z}) \geq I. \tag{4.13}$$

The information-rate function $I(R)$ allows us to quantify the maximum of the relevant information that can be preserved when the compression rate is at most R. Conversely, the rate-information function $R(I)$ quantifies the minimum compression rate required when the retained relevant information must be at least I. We note that the data processing inequality implies the following upper bound for $I(R)$:

$$I(R) \leq \min\{R, I(\mathbf{x};\mathbf{y})\}. \tag{4.14}$$

Figure 4.1 illustrates the information-rate function $I(R)$ (solid line) and the upper bound (4.14) (dashed lines). The shaded region in Figure 4.1 corresponds to the achievable rate-information pairs and, conversely, the hatched region corresponds to rate-information pairs that are not achievable. We discuss achievability for the Gaussian case in Section 4.6. An IB coding theorem for discrete random variables is given in [31].

The definition in (4.12) is structurally similar to the distortion-rate function, with the difference that the minimization of the distortion is replaced by a maximization of the relevant information. Analogously, (4.13) is structurally similar to the rate-distortion function, where the upper bound on the distortion is replaced by a lower bound on the relevant information. We emphasize that, contrary to RD theory, no distortion is involved in the definition of $I(R)$ and $R(I)$. Hence, the values of $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are immaterial (note that mutual information depends only on the probability distributions).

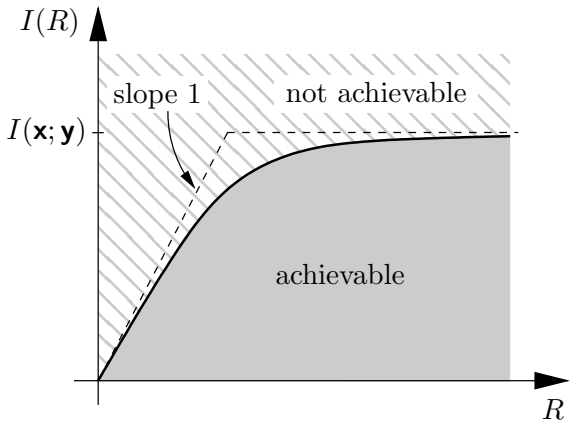


Figure 4.1: Illustration of the information-rate function $I(R)$.

4.3 The Gaussian Information Bottleneck

We briefly review the GIB [17] which we use in Sections 4.4 and 4.5 to derive closed-form expressions for the information-rate function and the rate-information function. For a Markov chain $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z}$ with jointly Gaussian $\mathsf{x} \in \mathbb{R}^m$ and $\mathsf{y} \in \mathbb{R}^n$, the GIB addresses the following variational problem:

$$\min_{p(\boldsymbol{z}|\boldsymbol{y})} I(\mathsf{y};\mathsf{z}) - \beta I(\mathsf{x};\mathsf{z}). \tag{4.15}$$

Like the IB problem (2.128), the problem in (4.15) considers the trade-off between compression rate $I(\mathsf{y};\mathsf{z})$ and relevant information $I(\mathsf{x};\mathsf{z})$ via the Lagrange parameter β. Here, the joint distribution of x and y is assumed to be known. As discussed in Section 4.1, we assume without loss of generality that x and y are zero-mean with full rank covariance matrices.

In [32] it has been shown that the optimal z solving (4.15) is jointly Gaussian with y and can therefore be written as

$$\mathsf{z} = \boldsymbol{A}\mathsf{y} + \boldsymbol{\xi}, \tag{4.16}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is a deterministic matrix and $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\xi}})$ is independent of y. Hence, we can rewrite the problem in (4.15) using (4.16) as

$$\min_{\boldsymbol{A},\boldsymbol{C}_{\boldsymbol{\xi}}} I(\mathsf{y};\boldsymbol{A}\mathsf{y}+\boldsymbol{\xi}) - \beta I(\mathsf{x};\boldsymbol{A}\mathsf{y}+\boldsymbol{\xi}). \tag{4.17}$$

Due to (4.16), the optimal $p(\boldsymbol{z}|\boldsymbol{y})$ and $p(\boldsymbol{z}|\boldsymbol{x}) = \int_{\mathbb{R}^n} p(\boldsymbol{z}|\boldsymbol{y})p(\boldsymbol{y}|\boldsymbol{x})\mathrm{d}\boldsymbol{y}$ are Gaussian distributions, too.

Denote by $\boldsymbol{v}_k^{\mathrm{T}}$ and λ_k, $k = 1,\ldots,n$, the left eigenvectors and associated eigenvalues of $\boldsymbol{C}_{\mathsf{y}|\boldsymbol{x}}\boldsymbol{C}_{\mathsf{y}}^{-1}$, where $\boldsymbol{C}_{\mathsf{y}} = \mathbb{E}\{\mathsf{y}\mathsf{y}^{\mathrm{T}}\}$ and $\boldsymbol{C}_{\mathsf{y}|\boldsymbol{x}} = \mathbb{E}\{\mathsf{y}\mathsf{y}^{\mathrm{T}}|\mathsf{x}{=}\boldsymbol{x}\}$ are, respectively, the unconditional and the conditional covariance matrix of y. An optimal solution of (4.17) can then be written as [17, Theorem 3.1]

$$\boldsymbol{A} = \mathrm{diag}\{\alpha_k\}_{k=1}^n \boldsymbol{V}^{\mathrm{T}} \quad \text{and} \quad \boldsymbol{C}_{\boldsymbol{\xi}} = \boldsymbol{I}, \tag{4.18}$$

where $\boldsymbol{V} = [\boldsymbol{v}_1 \cdots \boldsymbol{v}_n]$ and

$$\alpha_k = \sqrt{\frac{[\beta(1-\lambda_k) - 1]^+}{\lambda_k \boldsymbol{v}_k^{\mathrm{T}} \boldsymbol{C}_{\mathsf{y}} \boldsymbol{v}_k}}, \qquad k = 1, \ldots, n. \tag{4.19}$$

Using (4.18) and (4.19), the rate-information trade-off can implicitly be written as follows [17, Section 5]:

$$I(\mathbf{x}; \mathbf{z}) = I(\mathbf{y}; \mathbf{z}) - \frac{1}{2} \sum_{k=1}^{n} \log_2^+ \beta(1-\lambda_k). \tag{4.20}$$

The trade-off in (4.20) is parametrized by β. We note $I(\mathbf{x}; \mathbf{z})$ and $I(\mathbf{y}; \mathbf{z})$ are nondecreasing in β. Next, we shall use the implicit trade-off in (4.20) to derive the information-rate function and the rate-information function in closed form.

4.4 Scalar Case

In this section, we consider scalar jointly Gaussian random variables x and y. We treat the (univariate) scalar case and the (multivariate) vector case (cf. Section 4.5) separately, because the scalar case is easier to analyze and the results play an important role in vector case.

Using the linear model (4.1), we have

$$\mathsf{y} = h\mathsf{x} + \mathsf{w}, \tag{4.21}$$

where $h \in \mathbb{R}$ and w is independent of x. By properly choosing h and the variance of w, any joint distribution of x and y can be written as in (4.21). Specifically, let $\rho_{\mathsf{x},\mathsf{y}}$ denote the correlation coefficient of x and y, then we have $h = \rho_{\mathsf{x},\mathsf{y}} \sqrt{\operatorname{var}\{\mathsf{x}\}/\operatorname{var}\{\mathsf{y}\}}$ and $\operatorname{var}\{\mathsf{w}\} = \operatorname{var}\{\mathsf{y}\}(1 - \rho_{\mathsf{x},\mathsf{y}}^2)$. In the following we let $\mathsf{x} \sim \mathcal{N}(0, P)$ and $\mathsf{w} \sim \mathcal{N}(0, \sigma^2)$, yielding $\mathsf{y} \sim \mathcal{N}(0, h^2P + \sigma^2)$. We define the signal-to-noise ratio (SNR) of x and y as

$$\gamma \triangleq \frac{\rho_{\mathsf{x},\mathsf{y}}^2}{1 - \rho_{\mathsf{x},\mathsf{y}}^2} = \frac{h^2 P}{\sigma^2}. \tag{4.22}$$

Furthermore, we define

$$C(\gamma) \triangleq \frac{1}{2} \log_2(1 + \gamma). \tag{4.23}$$

We note that (4.23) is the capacity of a Gaussian channel with SNR γ under an average input power constraint [20, Section 9.1]. In the following theorem, we state a closed-form expression for the information-rate function and discuss its properties.

Theorem 4.3. *The information-rate function for jointly Gaussian random variables with SNR γ is given as*

$$I(R) = R - \frac{1}{2} \log_2 \frac{2^{2R} + \gamma}{1 + \gamma} \tag{4.24}$$

$$= C(\gamma) - C(2^{-2R}\gamma). \tag{4.25}$$

The information-rate function has the following properties:

1. $I(R)$ *is strictly concave on* $\mathbb{R}_+$.
2. $I(R)$ *is strictly increasing in* R.
3. $I(R) \leq \min\{R, C(\gamma)\}$.
4. $I(0) = 0$ *and* $\lim_{R\to\infty} I(R) = C(\gamma)$.
5. $\frac{\mathrm{d}I(R)}{\mathrm{d}R} = (1 + 2^{2R}\gamma^{-1})^{-1}$.

Proof: See Appendix B.1. ■

From (4.24) we conclude that $I(R) \approx R$ for small R. Similarly, (4.25) implies that $I(R) \approx C(\gamma)$ for large R. We call $R < C(\gamma)$ the *compression-limited* regime (since $\min\{R, C(\gamma)\} = R$) and we call $R > C(\gamma)$ the *noise-limited* regime (since $\min\{R, C(\gamma)\} = C(\gamma)$). Furthermore, we note that $\lim_{\gamma\to\infty} I(R) = R$. The following corollaries follow from Theorem 4.3.

Corollary 4.4. *Rate-information-optimal compression of* y *can be modeled as* $\mathsf{z} = \mathsf{y} + \mathsf{u}$, *where* u *is a zero-mean Gaussian random variable which is independent of* y *and has variance*

$$\sigma_{\mathsf{u}}^2 = \frac{\mathrm{var}\{\mathsf{y}\}}{2^{2R} - 1}. \tag{4.26}$$

We note that σ_{u}^2 *does not depend on the moments of* x.

Corollary 4.5. *The SNR of* x *and* z *equals*

$$\hat{\gamma} = \frac{\rho_{\mathsf{x},\mathsf{z}}^2}{1 - \rho_{\mathsf{x},\mathsf{z}}^2} = \gamma \frac{1 - 2^{-2R}}{1 + 2^{-2R}\gamma} \leq \gamma, \tag{4.27}$$

with the correlation coefficient $\rho_{\mathsf{x},\mathsf{z}} = \sqrt{(1 - 2^{-2R})\gamma/(1 + \gamma)}$.

Corollary 4.6. *The information-rate function can be written as* $I(R) = C(\hat{\gamma})$, *where* $\hat{\gamma}$ *is the SNR in (4.27).*

In a communications setting (where x and y are, respectively, the input and the output of a Gaussian channel with SNR γ), Theorem 4.3 has the following interpretation: The penalty (in terms of achievable data rate) for optimal quantization of the channel output y with rate R is asymptotically equal to $C(2^{-2R}\gamma)$. Hence, for any finite dimensional vector quantizer the achievable rate after quantization with rate R is reduced by *at least* $C(2^{-2R}\gamma)$.

Figure 4.2a depicts $I(R)$ versus R for different values of γ. The individual curves saturate at $C(\gamma)$ as R becomes large. For the same values of γ, we plot the SNR penalty $\Delta_\gamma(R) = \gamma/\hat{\gamma}$ (in dB) versus R in Figure 4.2b. These curves show the minimum R that is required to ensure that $\Delta_\gamma(R)$ is below a certain value. We observe that for fixed R, the SNR penalty increases with γ. In the following theorem, we give a closed-form expression for the rate-information function and discuss its properties.

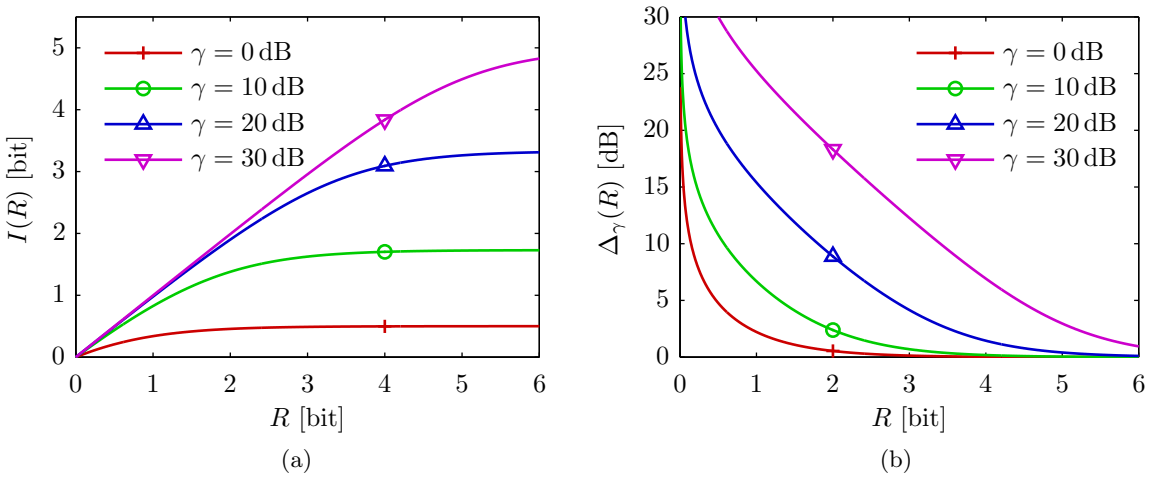


Figure 4.2: (a) Information-rate function and (b) SNR penalty (in dB) versus R for different values of γ.

Theorem 4.7. *The rate-information function for jointly Gaussian random variables with SNR γ is given as*

$$R(I) = \frac{1}{2}\log_2 \frac{\gamma}{2^{-2I}(1+\gamma)-1}. \tag{4.28}$$

The rate-information function has the following properties:

1. *$R(I)$ is strictly convex on $[0, C(\gamma)]$.*
2. *$R(I)$ is strictly increasing in I.*
3. *$R(I) \geq I$.*
4. *$R(0) = 0$ and $\lim_{I \to C(\gamma)} R(I) = \infty$.*
5. *$\frac{\mathrm{d}R(I)}{\mathrm{d}I} = (1+\gamma)/(1+\gamma-2^{2I})$.*

Proof: Rewriting the information-rate function directly yields (4.28). The proof of the properties of $R(I)$ is analogous to the proof of the properties of $I(R)$ in Appendix B.1. ■

Corollary 4.8. *The rate-information function (4.28) is the inverse of the information-rate function (4.24), i.e., for $\tilde{I} \in [0, C(\gamma)]$ and $\tilde{R} \in \mathbb{R}_+$ we have*

$$I(R(\tilde{I})) = \tilde{I} \qquad \text{and} \qquad R(I(\tilde{R})) = \tilde{R}. \tag{4.29}$$

Thus, the derivatives of $I(R)$ and $R(I)$ are related as

$$I'(\tilde{R}) = \frac{1}{R'\big(I(\tilde{R})\big)} \qquad \text{and} \qquad R'(\tilde{I}) = \frac{1}{I'\big(R(\tilde{I})\big)}. \tag{4.30}$$

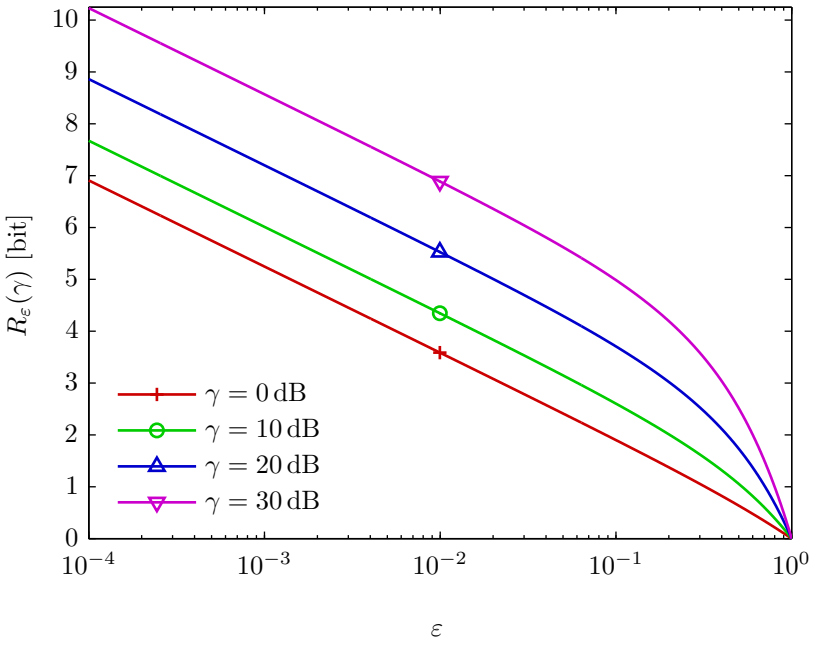


Figure 4.3: Minimum compression rate required to achieve $I(R) \geq (1-\varepsilon)C(\gamma)$.

Corollary 4.9. *Let $\varepsilon > 0$. The minimum compression rate $R_\varepsilon(\gamma)$ required to achieve $I(R) \geq (1-\varepsilon)C(\gamma)$ equals*

$$R_\varepsilon(\gamma) = \frac{1}{2}\log_2 \frac{\gamma}{(1+\gamma)^\varepsilon - 1}. \tag{4.31}$$

In Figure 4.3, we show $R_\varepsilon(\gamma)$ versus ε for different values of γ. We note that $R_\varepsilon(\gamma)$ increases exponentially as $\varepsilon \to 0$. For fixed ε, i.e., for a fixed gap to $C(\gamma)$, we observe that $R_\varepsilon(\gamma)$ increases with γ. For asymptotically low SNR, we have $\lim_{\gamma\to 0} R_\varepsilon(\gamma) = \log_2(1/\varepsilon)/2$.

4.5 Vector Case

We consider the linear model

$$\mathsf{y} = \boldsymbol{H}\mathsf{x} + \mathsf{w}, \tag{4.32}$$

with $\boldsymbol{H} \in \mathbb{R}^{n\times m}$, $\mathsf{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_\mathsf{x})$, and $\mathsf{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_\mathsf{w})$ independent of x. Due to (4.32), the covariance matrix of y is given as

$$\boldsymbol{C}_\mathsf{y} = \boldsymbol{H}\boldsymbol{C}_\mathsf{x}\boldsymbol{H}^\mathrm{T} + \boldsymbol{C}_\mathsf{w}. \tag{4.33}$$

Let $\boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{U}^\mathrm{T}$ denote the eigendecomposition of the positive semidefinite matrix $\boldsymbol{C}_\mathsf{w}^{-1/2}\boldsymbol{H}\boldsymbol{C}_\mathsf{x}\boldsymbol{H}^\mathrm{T}\boldsymbol{C}_\mathsf{w}^{-1/2}$, where $\boldsymbol{U}$ is an orthogonal matrix and $\boldsymbol{\Gamma} = \mathrm{diag}\{\gamma_k\}_{k=1}^n$ is a diagonal matrix of nonnegative eigenvalues γ_k, $k = 1, \ldots, n$. In what follows, we work with the

whitened and rotated random vector

$$\widetilde{\mathbf{y}} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{C}_{\mathbf{w}}^{-1/2}\mathbf{y} = \widetilde{\mathbf{x}} + \widetilde{\mathbf{w}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma} + \boldsymbol{I}). \tag{4.34}$$

The transformation $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{C}_{\mathbf{w}}^{-1/2}$ simultaneously diagonalizes the "signal" covariance $\boldsymbol{C}_{\mathbf{x}}$ and the "noise" covariance $\boldsymbol{C}_{\mathbf{w}}$, i.e., we have $\widetilde{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ and $\widetilde{\mathbf{w}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Therefore, the transformation (4.34) decouples the linear model (4.32) into n independent *modes*

$$\widetilde{\mathsf{y}}_k = \widetilde{\mathsf{x}}_k + \widetilde{\mathsf{w}}_k, \qquad k = 1, \ldots, n, \tag{4.35}$$

with mode SNRs $\gamma_k = \rho^2_{\widetilde{\mathsf{x}}_k, \widetilde{\mathsf{y}}_k}/(1 - \rho^2_{\widetilde{\mathsf{x}}_k, \widetilde{\mathsf{y}}_k})$. We note that (4.34) is an invertible transformation and therefore does not affect mutual information. Due to (4.35) we expect the results in the vector case to be structurally similar to the results in the scalar case.

Without loss of generality we assume in the following that the mode SNRs are sorted in descending order, i.e., we have $\gamma_1 \geq \cdots \geq \gamma_n$. The following theorem gives a closed-form expression for the information-rate function in the vector case and discusses its properties.

Theorem 4.10. *The information-rate function for jointly Gaussian random vectors with sorted mode SNRs γ_k, $k = 1, \ldots, n$, is given as*

$$I(R) = R - \frac{1}{2}\sum_{k=1}^{n} \log_2 \frac{2^{2R_k(R)} + \gamma_k}{1 + \gamma_k} \tag{4.36}$$

$$= \sum_{k=1}^{n} \big(C(\gamma_k) - C(2^{-2R_k(R)}\gamma_k)\big), \tag{4.37}$$

where the compression rate allocated to the kth mode equals

$$R_k(R) = \begin{cases} \left[\dfrac{R}{\ell(R)} + \dfrac{1}{2}\log_2 \dfrac{\gamma_k}{\prod_{i=1}^{\ell(R)} \gamma_i^{1/\ell(R)}}\right]^+, & R > 0 \\ 0, & R = 0 \end{cases}. \tag{4.38}$$

Here, $\ell(R)$ denotes the number of active modes which is given as

$$\ell(R) = \sum_{k=1}^{n} \mathbb{1}\{R > R_{c,k}\}, \tag{4.39}$$

where

$$R_{c,k} = \frac{1}{2}\sum_{i=1}^{k} \log_2 \frac{\gamma_i}{\gamma_k}, \quad k = 1, \ldots, n, \tag{4.40}$$

are the critical rates. The information-rate function has the following properties:

1. *$I(R)$ is strictly concave on $\mathbb{R}_+$.*
2. *$I(R)$ is strictly increasing in R.*

3. $I(R) \leq \min\{R, \sum_{k=1}^{n} C(\gamma_k)\}$.

4. $I(0) = 0$ *and* $\lim_{R\to\infty} I(R) = \sum_{k=1}^{n} C(\gamma_k)$.

5. $\frac{\mathrm{d}}{\mathrm{d}R} I(R) = \frac{1}{\ell(R)} \sum_{k=1}^{\ell(R)} (1 + 2^{2R_k(R)}\gamma_k^{-1})^{-1}$.

Proof: See Appendix B.2. ■

We note that the information-rate function (4.36) is the sum of n information-rate functions for scalar jointly Gaussian random variables with SNRs γ_k, $k = 1, \ldots, n$ (cf. (4.24)). The number of active modes, i.e., the number of modes with $R_k(R) > 0$, increases at the critical rates. More precisely, for any $\varepsilon > 0$ we have $\ell(R_{c,k} + \varepsilon) > \ell(R_{c,k})$, $k = 1, \ldots, n$. Note that $R_{c,1} = 0$ and thus $I(R) > 0$ for $R > 0$. In Section 4.6, we shall show that the optimal rate allocation (4.38) is obtained by performing reverse waterfilling on the mode SNRs. The following corollaries are consequences of Theorem 4.10.

Corollary 4.11. *Rate-information-optimal compression of* $\widetilde{\mathbf{y}}$ *can be modeled as* $\widetilde{\mathbf{z}} = \widetilde{\mathbf{y}} + \mathbf{u}$, *with* $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_{\mathbf{u}})$ *independent of* $\widetilde{\mathbf{y}}$ *and*

$$\boldsymbol{C}_{\mathbf{u}} = \operatorname{diag}\left\{ \frac{1+\gamma_k}{2^{2R_k(R)} - 1} \right\}_{k=1}^{n}. \tag{4.41}$$

Corollary 4.12. *The SNR of* $\widetilde{\mathsf{x}}_k$ *and* $\widetilde{\mathsf{z}}_k$, *i.e., the SNR of the* k*th mode after compression, equals*

$$\hat{\gamma}_k = \frac{\rho^2_{\widetilde{\mathsf{x}}_k, \widetilde{\mathsf{z}}_k}}{1 - \rho^2_{\widetilde{\mathsf{x}}_k, \widetilde{\mathsf{z}}_k}} = \gamma_k \frac{1 - 2^{-2R_k(R)}}{1 + 2^{-2R_k(R)}\gamma_k} \leq \gamma_k, \tag{4.42}$$

with the correlation coefficient $\rho_{\widetilde{\mathsf{x}}_k, \widetilde{\mathsf{z}}_k} = \sqrt{(1 - 2^{-2R_k(R)})\gamma_k/(1+\gamma_k)}$.

Corollary 4.13. *The information-rate function can be written as* $I(R) = \sum_{k=1}^{n} C(\hat{\gamma}_k)$, *where* $\hat{\gamma}_k$, $k = 1, \ldots, n$, *are the SNRs in (4.42).*

In the following theorem, we state the rate-information function and its properties.

Theorem 4.14. *The rate-information function for jointly Gaussian random vectors with sorted mode SNRs* γ_k, $k = 1, \ldots, n$, *is given as*

$$R(I) = \frac{1}{2} \sum_{k=1}^{n} \log_2 \frac{\gamma_k}{2^{-2I_k(I)}(1+\gamma_k) - 1}, \tag{4.43}$$

where the relevant information of the k*th mode equals*

$$I_k(I) = \begin{cases} \left[\dfrac{I}{\ell(I)} + \dfrac{1}{2} \log_2 \dfrac{1+\gamma_k}{\prod_{i=1}^{\ell(I)} (1+\gamma_i)^{1/\ell(I)}} \right]^+, & I > 0 \\ 0, & I = 0 \end{cases}. \tag{4.44}$$

Here, $\ell(I)$ denotes the number of active modes which is given as

$$\ell(I) = \sum_{k=1}^{n} \mathbb{1}\{I > I_{c,k}\}, \tag{4.45}$$

where

$$I_{c,k} = \frac{1}{2} \sum_{i=1}^{k} \log_2 \frac{1+\gamma_i}{1+\gamma_k}, \quad k = 1, \ldots, n, \tag{4.46}$$

are the critical values of the relevant information. The rate-information function has the following properties:

1. *$R(I)$ is strictly convex on $\left[0, \sum_{k=1}^{n} C(\gamma_k)\right]$.*
2. *$R(I)$ is strictly increasing in I.*
3. *$R(I) \geq I$.*
4. *$R(0) = 0$ and $\lim_{I \to \sum_{k=1}^{n} C(\gamma_k)} R(I) = \infty$.*
5. *$\frac{\mathrm{d}}{\mathrm{d}I} R(I) = \frac{1}{\ell(I)} \sum_{k=1}^{\ell(I)} (1+\gamma_k)/(1+\gamma_k - 2^{2I_k(I)})$.*

Proof: See Appendix B.3. ∎

The rate-information function (4.43) can be identified as the sum of n rate-information functions of scalar Gaussian channels with SNRs γ_k, $k = 1, \ldots, n$, (cf. (4.28)). The number of active modes increases at the critical values of the relevant information. More precisely, for any $\varepsilon > 0$ we have $\ell(I_{c,k} + \varepsilon) > \ell(I_{c,k})$, $k = 1, \ldots, n$. Note that $I_{c,1} = 0$ and thus $R(I) > 0$ for $I > 0$. The following corollaries follow from the properties of $I(R)$ and $R(I)$.

Corollary 4.15. *The rate-information function (4.43) is the inverse of the information-rate function (4.36), i.e., for $\tilde{I} \in \left[0, \sum_{k=1}^{n} C(\gamma_k)\right]$ and $\tilde{R} \in \mathbb{R}_+$ we have*

$$I(R(\tilde{I})) = \tilde{I} \qquad \text{and} \qquad R(I(\tilde{R})) = \tilde{R}. \tag{4.47}$$

Thus, the derivatives of $I(R)$ and $R(I)$ are related as

$$I'(\tilde{R}) = \frac{1}{R'\big(I(\tilde{R})\big)} \qquad \text{and} \qquad R'(\tilde{I}) = \frac{1}{I'\big(R(\tilde{I})\big)}. \tag{4.48}$$

Corollary 4.16. *Let $\varepsilon > 0$. The minimum compression rate $R_\varepsilon(\gamma_1, \ldots, \gamma_n)$ required to achieve $I(R) \geq I_\varepsilon$, with $I_\varepsilon = (1-\varepsilon) \sum_{k=1}^{n} C(\gamma_k)$, equals*

$$R_\varepsilon(\gamma_1, \ldots, \gamma_n) = \frac{1}{2} \sum_{k=1}^{\ell(I_\varepsilon)} \log_2 \frac{\gamma_k}{\prod_{k=1}^{\ell(I_\varepsilon)} (1+\gamma_k)^{\frac{\varepsilon}{\ell(I_\varepsilon)}} \prod_{k=\ell(I_\varepsilon)+1}^{n} (1+\gamma_k)^{-\frac{1-\varepsilon}{\ell(I_\varepsilon)}} - 1}. \tag{4.49}$$

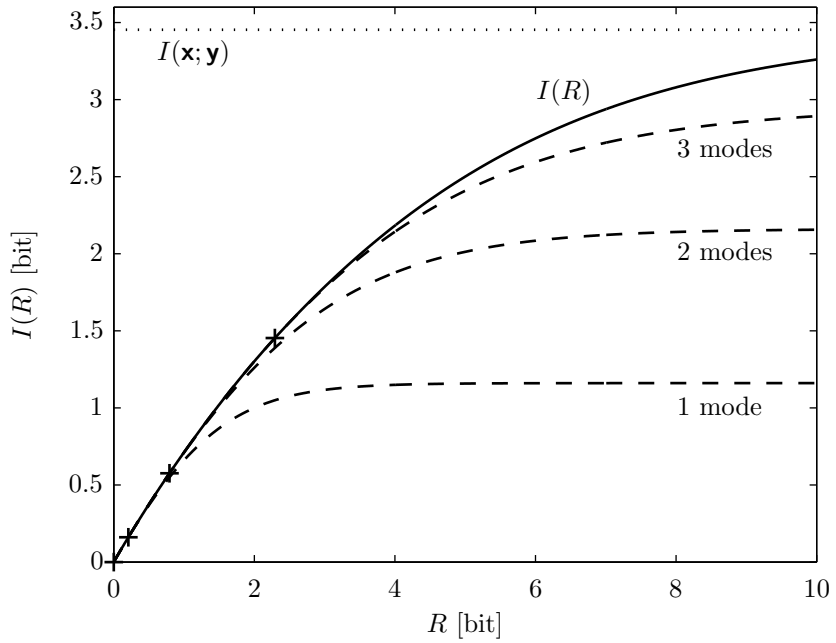


Figure 4.4: Phase transitions at the critical rates in the rate-information trade-off. In this example, the mode SNRs are given as $\gamma_k = 5 - k$, $k = 1, \ldots, 4$.

We note that the results in the vector case are structurally equivalent to those in the scalar case. Therefore, we refer to Figures 4.2 and 4.3 for an illustration of our results. The major difference is the optimal allocation of the compression rate to the individual modes in the vector case. Figure 4.4 shows the phase transitions in the rate-information trade-off that occur at the critical rates $R_{c,k}$ (indicated by '+' markers) for $\gamma_k = 5-k$, $k = 1, \ldots, 4$. The solid curve shows $I(R)$ and the dashed curves show the information-rate functions corresponding to subsets of the modes. Specifically, the curve labeled "k mode(s)" is the information-rate function corresponding to the k strongest modes with SNRs $\gamma_1, \ldots, \gamma_k$. Furthermore, this curve is equal to $I(R)$ for rates below the critical rate $R_{c,k+1}$ and bifurcates at $R_{c,k+1}$. The dotted line in Figure 4.4 corresponds to $I(\mathbf{x};\mathbf{y}) = \lim_{R\to\infty} I(R)$.

4.6 Connections to Rate-Distortion Theory

In this section, we compare rate-information-optimal compression to RD-optimal compression with squared-error distortion. We first study the rate-information trade-off achievable using MSE-optimal quantization. To make our analysis more flexible, we consider compression of a linearly filtered version of $\widetilde{\mathbf{y}}$ (cf. (4.34)). Specifically, we let

$$\check{\mathbf{y}} = \boldsymbol{F}\widetilde{\mathbf{y}} = \boldsymbol{F}(\widetilde{\mathbf{x}} + \widetilde{\mathbf{w}}), \tag{4.50}$$

where $\boldsymbol{F} = \mathrm{diag}\{f_k\}_{k=1}^n$, and we optimally compress $\check{\mathbf{y}}$ in the RD sense using the squared-error distortion measure. We denote the resulting rate-information trade-off by $I^{\mathrm{RD}}(R, \boldsymbol{F})$. Note that the optimality of the GIB implies $I(R) \geq I^{\mathrm{RD}}(R, \boldsymbol{F})$. The special cases of MSE-optimal source coding and MSE-optimal noisy source coding correspond to $\boldsymbol{F} = \boldsymbol{I}$ and $\boldsymbol{F} = \boldsymbol{W}$, respectively. Here, $\boldsymbol{W}$ is the Wiener filter for estimating $\widetilde{\mathbf{x}}$ from $\widetilde{\mathbf{y}}$, i.e., we have

$$\boldsymbol{W} = \boldsymbol{\Gamma}(\boldsymbol{I} + \boldsymbol{\Gamma})^{-1}. \tag{4.51}$$

We next give a closed-form expression of $I^{\mathrm{RD}}(R, \boldsymbol{F})$ and discuss its consequences.

Lemma 4.17. *The rate-information trade-off achievable by RD-optimal compression of* $\check{\mathbf{y}} = \boldsymbol{F}\widetilde{\mathbf{y}}$ *with squared-error distortion, where* $\widetilde{\mathbf{y}} \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}\{1+\gamma_k\}_{k=1}^n)$ *and* $\boldsymbol{F} = \mathrm{diag}\{f_k\}_{k=1}^n$*, is given as*

$$I^{\mathrm{RD}}(R, \boldsymbol{F}) = \frac{1}{2}\sum_{k=1}^{n} \log_2 \frac{1+\gamma_k}{1 + 2^{-2R_k^{\mathrm{RD}}(R,\boldsymbol{F})}\gamma_k}. \tag{4.52}$$

With $\omega_k \triangleq f_k^2(1+\gamma_k)$*,* $k = 1, \ldots, n$*, sorted in descending order, the compression rate allocated to the* k*th mode equals*

$$R_k^{\mathrm{RD}}(R, \boldsymbol{F}) = \begin{cases} \left[\dfrac{R}{l(R,\boldsymbol{F})} + \dfrac{1}{2}\log_2 \dfrac{\omega_k}{\prod_{i=1}^{l(R,\boldsymbol{F})} \omega_i^{1/l(R,\boldsymbol{F})}}\right]^+, & R > 0 \\ 0, & R = 0 \end{cases}. \tag{4.53}$$

Here, $l(R, \boldsymbol{F})$ *denotes the number of active modes which is given as*

$$l(R, \boldsymbol{F}) = \sum_{k=1}^{n} \mathbb{1}\left\{R > R_{c,k}^{\mathrm{RD}}(\boldsymbol{F})\right\}, \tag{4.54}$$

where

$$R_{c,k}^{\mathrm{RD}}(\boldsymbol{F}) = \frac{1}{2}\sum_{i=1}^{k} \log_2 \frac{\omega_i}{\omega_k}, \quad k = 1, \ldots, n, \tag{4.55}$$

are the critical rates.

Proof: See Appendix B.4. ∎

We note that $I^{\mathrm{RD}}(R, \boldsymbol{F})$ is invariant with respect to scaling of $\boldsymbol{F}$, i.e., we have $I^{\mathrm{RD}}(R, \boldsymbol{F}) = I^{\mathrm{RD}}(R, \alpha\boldsymbol{F})$ for any $\alpha \neq 0$. Furthermore, the only difference between $I^{\mathrm{RD}}(R, \boldsymbol{F})$ and $I(R)$ is the rate allocation. Indeed, we can write $I(R)$ (cf. (4.36)) as

$$I(R) = \frac{1}{2}\sum_{k=1}^{n} \log_2 \frac{1+\gamma_k}{1 + 2^{-2R_k(R)}\gamma_k}. \tag{4.56}$$

Thus, an obvious question is whether there exists an $\boldsymbol{F}$ such that $R_k^{\mathrm{RD}}(R, \boldsymbol{F}) = R_k(R)$ for all $R \in \mathbb{R}_+$. The following theorem answers this question in the affirmative.

Theorem 4.18. *The optimal rate-information trade-off for jointly Gaussian random vectors with mode SNRs γ_k, $k = 1, \ldots, n$, can be achieved by linear filtering with subsequent MSE-optimal source coding. Specifically, we have*

$$I(R) = \max_{\boldsymbol{F}} I^{\mathrm{RD}}(R, \boldsymbol{F}) = I^{\mathrm{RD}}(R, \boldsymbol{F}^\star), \tag{4.57}$$

with an optimal linear filter

$$\boldsymbol{F}^\star = \arg\max_{\boldsymbol{F}} I^{\mathrm{RD}}(R, \boldsymbol{F}) = \operatorname{diag}\left\{\sqrt{\frac{\gamma_k}{1+\gamma_k}}\right\}_{k=1}^{n}. \tag{4.58}$$

The solution of (4.57) is not unique since any $\boldsymbol{F}_\alpha^\star = \alpha \boldsymbol{F}^\star$ with $\alpha \neq 0$ is optimal. We identify $\boldsymbol{F}^\star$ as the positive square root of the Wiener filter for a Gaussian signal in noise problem.

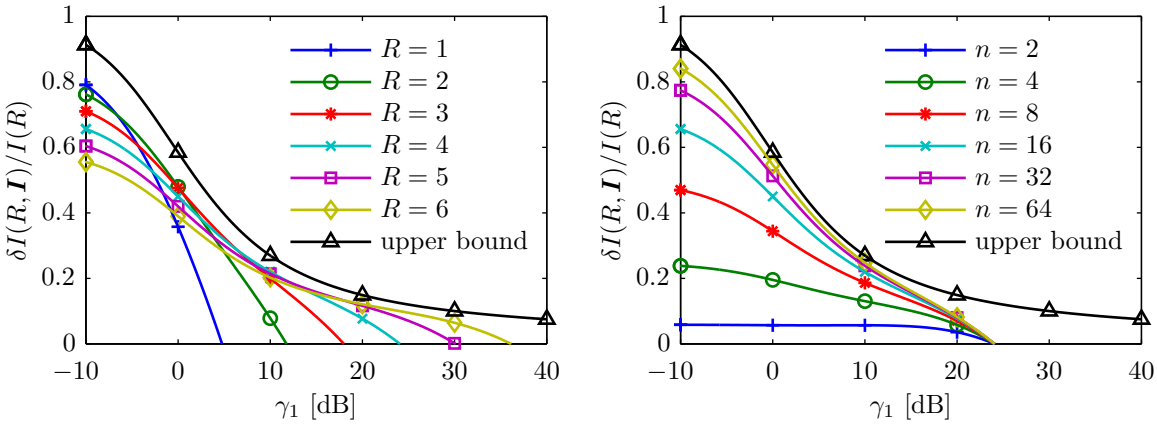
Proof: Using Lemma 4.17, it is not hard to see that (4.52)-(4.55) is equal to the optimal rate-information trade-off (4.36)-(4.40) if $f_k = \alpha\sqrt{\gamma_k/(1+\gamma_k)}$ for any $\alpha \neq 0$. ■

As a consequence of Theorem 4.18, RD theory provides achievability and converse results for the rate-information trade-off. Hence, there exist codes which asymptotically achieve the optimal trade-off and $I(R)$ is indeed the dividing line between what is achievable and what is not. Furthermore, Theorem 4.18 implies that the GIB can be decomposed into linear filtering and MSE-optimal source coding. This is convenient since linear systems and RD theory are very well understood and more widely known than the GIB.

Since the optimal linear filter $\boldsymbol{F}$ is the square-root Wiener filter $\boldsymbol{F}^\star = \boldsymbol{W}^{1/2}$, we conclude that both MSE-optimal source coding (corresponding to $\boldsymbol{F} = \boldsymbol{I}$) and MSE-optimal noisy source coding (corresponding to $\boldsymbol{F} = \boldsymbol{W}$) are suboptimal in general. For an arbitrary $\boldsymbol{F} = \operatorname{diag}\{f_k\}_{k=1}^n$, we can express and upper bound the gap to the optimal rate-information trade-off as follows:

$$\begin{aligned}
\delta I(R, \boldsymbol{F}) &\triangleq I(R) - I^{\mathrm{RD}}(R, \boldsymbol{F}) && (4.59)\\
&= \frac{1}{2}\sum_{k=1}^{n} \log_2 \frac{1 + 2^{-2R_k^{\mathrm{RD}}(R,\boldsymbol{F})}\gamma_k}{1 + 2^{-2R_k(R)}\gamma_k} && (4.60)\\
&\leq \sum_{k=1}^{n} C(\gamma_k) - \frac{1}{2}\log_2 \frac{f_1^2(1+\gamma_1)^2}{f_1^2(1+\gamma_1) + f_2^2\gamma_1(1+\gamma_2)}. && (4.61)
\end{aligned}$$

Clearly, we have $\delta I(R, \boldsymbol{F}^\star) = 0$. Moreover, if all nonzero mode SNRs are equal, i.e., if $\gamma_k \in \{\gamma, 0\}$, $k = 1, \ldots, n$, then we have $\delta I(R, \boldsymbol{F}) = 0$ for any linear filter with $f_k = \alpha \mathbb{1}\{\gamma_k \neq 0\}$ and $\alpha \neq 0$. In this case, MSE-optimal noisy source coding is rate-information-optimal, i.e., we have $I^{\mathrm{RD}}(R, \boldsymbol{W}) = I(R)$. Moreover, if all mode SNRs are nonzero and equal then MSE-optimal source coding is rate-information-optimal, too. In this case we have $I^{\mathrm{RD}}(R, \boldsymbol{I}) = I(R)$. In particular, MSE-optimal processing is always rate-information-optimal in the scalar case. We conclude that the gap $\delta I(R, \boldsymbol{F})$ can be expected to be large if the mode SNRs differ widely.



(a) (b)

Figure 4.5: (a) $\delta I(R,\boldsymbol{I})/I(R)$ versus γ_1 for different R and $n=16$. (b) $\delta I(R,\boldsymbol{I})/I(R)$ versus γ_1 for different n and $R=4$ bit.

Next, we explicitly analyze the gap $\delta I(R,\boldsymbol{F})$ for $\boldsymbol{F}=\boldsymbol{I}$ and $\gamma_1>0$, $\gamma_2=\cdots=\gamma_n=0$. In this case, the RD-optimal approach allocates rate to *all* modes if $R>R_{c,2}^{\mathrm{RD}}(\boldsymbol{I})=C(\gamma_1)$. Evaluating (4.60) and (4.61) yields

$$\delta I(R,\boldsymbol{I})=\begin{cases}0, & R\le R_{c,2}^{\mathrm{RD}}\\ \dfrac{1}{2}\log_2\dfrac{1+2^{-2R/n}\gamma_1(1+\gamma_1)^{1/n-1}}{1+2^{-2R}\gamma_1}, & R>R_{c,2}^{\mathrm{RD}}\end{cases} \tag{4.62}$$

$$\le C\left(\frac{\gamma_1}{1+\gamma_1}\right). \tag{4.63}$$

Figure 4.5a shows (4.62) and (4.63) normalized by $I(R)$ versus γ_1 for different rates and $n=16$. We observe that at low SNR $\delta I(R,\boldsymbol{I})/I(R)$ decreases with increasing R. Figure 4.5b shows (4.62) and (4.63) normalized by $I(R)$ versus γ_1 for different n and $R=4$ bit. Here, the upper bound (4.63) gets tighter as n increases. In Figure 4.5a and Figure 4.5b, we have $\delta I(R,\boldsymbol{I})/I(R)=0$ when γ_1 is such that $C(\gamma_1)\ge R$.

The proof of Lemma 4.17 shows that $I^{\mathrm{RD}}(R,\boldsymbol{F})$ admits an implicit reverse waterfilling representation. Specifically, we have

$$R(\theta,\boldsymbol{F})=\frac{1}{2}\sum_{k=1}^{n}\log_2^+\frac{\omega_k}{\theta}, \tag{4.64}$$

$$I^{\mathrm{RD}}(\theta,\boldsymbol{F})=\frac{1}{2}\sum_{k=1}^{n}\log_2^+\frac{1+\gamma_k}{1+\theta\gamma_k/\omega_k}, \tag{4.65}$$

where $\theta>0$ is the waterlevel. A waterfilling formulation of $I(R)$ is obtained by letting $\omega_k=\gamma_k$ in (4.64) and (4.65). The optimal rate allocation (4.38) therefore corresponds to

reverse waterfilling on the mode SNRs γ_k, $k = 1, \ldots, n$, i.e., we have

$$R_k(\theta) = \frac{1}{2} \log_2^+ \frac{\gamma_k}{\theta}, \tag{4.66}$$

where the waterlevel θ is chosen such that $\sum_{k=1}^n R_k(\theta) = R$. Indeed, it can be shown that (4.66) is the solution of the following convex optimization problem (cf., e.g., [12, Section 5.5]):

$$\begin{aligned} &\max_{R_1,\ldots,R_n} \frac{1}{2} \sum_{k=1}^n \log_2 \frac{1+\gamma_k}{1+2^{-2R_k}\gamma_k} \\ &\text{subject to} \quad \sum_{k=1}^n R_k = R, \quad R_k \geq 0, \quad k = 1, \ldots, n. \end{aligned} \tag{4.67}$$

The following Lemma considers properties of $I^{\mathrm{RD}}(R, \boldsymbol{F})$ and its derivative.

Lemma 4.19. *Let the quantities $\omega_k = f_k^2(1+\gamma_k)$, $k = 1, \ldots, n$, be sorted in descending order. The rate-information trade-off $I^{\mathrm{RD}}(R, \boldsymbol{F})$ is concave in R for arbitrary mode SNRs γ_k, $k = 1, \ldots, n$, if and only if the linear filter $\boldsymbol{F} = \operatorname{diag}\{f_k\}_{k=1}^n$ is such that*

$$\frac{\omega_{k+1}}{\omega_k} \geq \frac{\gamma_{k+1}}{\gamma_k}, \qquad k = 1, \ldots, n-1. \tag{4.68}$$

In particular, for nonnegative ρ, $I^{\mathrm{RD}}(R, \boldsymbol{W}^\rho)$ is concave in R for arbitrary mode SNRs if and only if $\rho \leq 1/2$. Furthermore, the derivative $\mathrm{d}I^{\mathrm{RD}}(R, \boldsymbol{F})/\mathrm{d}R$ is continuous, nonincreasing, and convex in R for arbitrary mode SNRs if and only if $\boldsymbol{F} = \boldsymbol{F}^\star$. Otherwise, i.e., if $\boldsymbol{F} \neq \boldsymbol{F}^\star$, $\mathrm{d}I^{\mathrm{RD}}(R, \boldsymbol{F})/\mathrm{d}R$ is discontinuous at the critical rates $R_{c,k}^{\mathrm{RD}}(\boldsymbol{F}) = \frac{1}{2}\sum_{i=1}^k \log_2 \frac{\omega_i}{\omega_k}$, $k = 2, \ldots, n$.

Proof: See Appendix B.5. ∎

Since (4.68) is fulfilled when $\boldsymbol{F} = \boldsymbol{F}^\star$, Lemma 4.19 implies that $I(R)$ is strictly increasing and concave and thus $R(I)$ is strictly increasing and convex. Furthermore, Lemma 4.19 shows that $I^{\mathrm{RD}}(R, \boldsymbol{W})$ is not concave in general and must therefore be suboptimal. In the following result, we state an interesting connection between the optimal critical rates $R_{c,k}$ and the critical rates $R_{c,k}^{\mathrm{RD}}(\boldsymbol{F})$.

Lemma 4.20. *The critical rates $R_{c,k}^{\mathrm{RD}}(\boldsymbol{I})$, $R_{c,k}^{\mathrm{RD}}(\boldsymbol{F}^\star)$, and $R_{c,k}^{\mathrm{RD}}(\boldsymbol{W})$ are related as follows:*

$$R_{c,k} = R_{c,k}^{\mathrm{RD}}(\boldsymbol{F}^\star) = \frac{R_{c,k}^{\mathrm{RD}}(\boldsymbol{I}) + R_{c,k}^{\mathrm{RD}}(\boldsymbol{W})}{2}, \quad k = 1, \ldots, n. \tag{4.69}$$

Furthermore, the critical rates are ordered such that

$$R_{c,k}^{\mathrm{RD}}(\boldsymbol{I}) \leq R_{c,k} \leq R_{c,k}^{\mathrm{RD}}(\boldsymbol{W}), \quad k = 1, \ldots, n, \tag{4.70}$$

which implies the following ordering of the number of active modes:

$$l(R, \boldsymbol{I}) \geq \ell(R) \geq l(R, \boldsymbol{W}). \tag{4.71}$$

Proof: See Appendix B.6. ■

Lemma 4.20 shows that MSE-optimal source coding uses too many modes and allocates too little rate to the strongest modes. Similarly, MSE-optimal noisy source coding uses too few modes and allocates too much rate to the strongest modes. Interestingly, the optimal critical rates $R_{c,k}$ are equal to the arithmetic mean of $R_{c,k}^{\mathrm{RD}}(\boldsymbol{I})$ and $R_{c,k}^{\mathrm{RD}}(\boldsymbol{W})$.

4.7 Quantizer Design

We next compare the asymptotic limit characterized by $I(R)$ to the relevant information that can be preserved using finite blocklength quantizers. To this end, we propose to quantize the modes with an MSE-optimal quantizer. Due to Theorem 4.18, we know that this strategy is asymptotically rate-information-optimal when suitable linear preprocessing is performed. In the finite blocklength regime, the rate-information-optimality of MSE-optimal quantization is not guaranteed. However, the numerical results presented below justify our approach.

An important property of MSE-optimal quantizers is that their quantization regions are disjoint convex sets (cf. Section 5.2). We note that convex quantization regions are not always optimal in quantizer design for communication problems (see, e.g., [120] for a counterexample). The existence of an MSE-optimal partition of the input space implies that randomized quantization cannot improve upon deterministic quantization. This is because any randomized quantizer can be realized by using a set of (possibly suboptimal) deterministic quantizers in a time-sharing manner.

The fact that we can restrict our attention to MSE-optimal quantizers is very convenient with respect to quantizer design. Specifically, MSE-optimal quantizers can be designed using well-known algorithms such as the Lloyd-Max algorithm [66, 71] and the LBG algorithm [65]. For the case of a single mode with $\gamma \in \{0\,\mathrm{dB}, 5\,\mathrm{dB}, 10\,\mathrm{dB}\}$, Figure 4.6 shows how close we can get to $I(R)$ using scalar quantizers. The solid lines correspond to the respective information-rate functions and the '$\times$' markers correspond to the relevant information achievable using MSE-optimal quantizers with 2 to 32 quantization levels. We note that in this case the quantization rate equals the entropy of the quantizer output, i.e., $R = I(\mathsf{y};\mathsf{z}) = H(\mathsf{z})$, since the quantizers are deterministic. We observe that the gap to $I(R)$ decreases as R increases and for fixed R the gap to $I(R)$ grows with increasing SNR. Using a vector quantizer instead of a scalar quantizer will slightly reduce the gap to $I(R)$. However, the main benefit of vector quantization (VQ) is the increased flexibility regarding the rate R. We note that time-sharing can be used to (asymptotically) achieve all points on a line connecting the rate-information pairs corresponding to two quantizers.

In the vector case with multiple modes we can design MSE-optimal vector quantizers which jointly quantize all modes. To ensure the correct rate allocation, the input of the vector quantizer has to be linearly filtered as described in Theorem 4.18. Alternatively, we may quantize the modes separately with a different MSE-optimal quantizer for each mode. In this case, linear preprocessing of the modes is not required but the rates of the individual

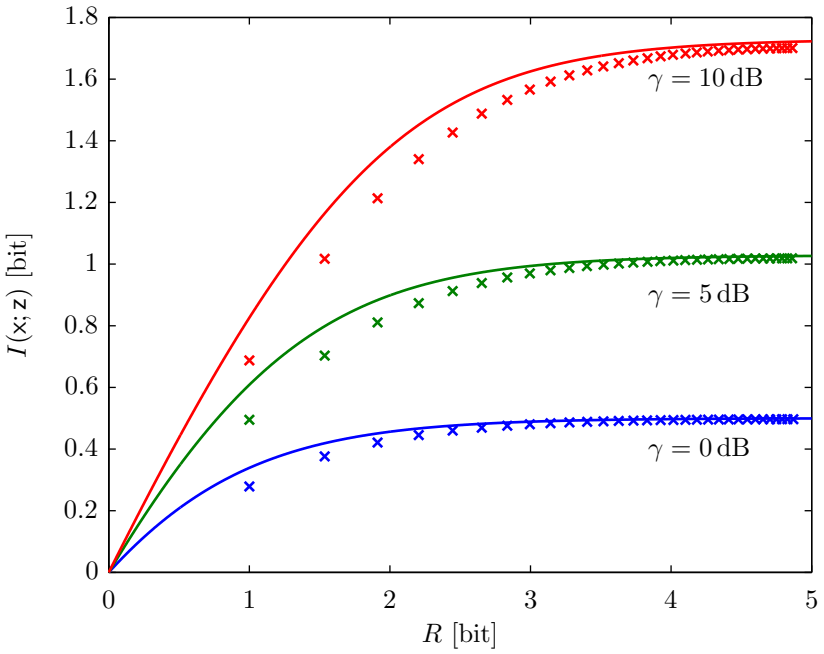


Figure 4.6: Comparison of $I(R)$ to the relevant information $I(\mathsf{x};\mathsf{z})$ achievable using scalar quantizers with 2 to 32 quantization levels for a single mode with $\gamma \in \{0\,\mathrm{dB}, 5\,\mathrm{dB}, 10\,\mathrm{dB}\}$.

quantizers must be chosen to closely match the optimal rate allocation. However, using quantizers with small blocklength it is hardly possible to obtain a good approximation of the optimal rate allocation. Therefore, the simpler strategy of separate quantization of the modes can be expected to perform worse than joint VQ of all modes.

Next, we give a numerical justification for the MSE as optimality criterion in the quantizer design. We again consider the case of a single mode with $\gamma \in \{0\,\mathrm{dB}, 5\,\mathrm{dB}, 10\,\mathrm{dB}\}$. Since the input of the quantizer is a zero-mean Gaussian distribution, the MSE-optimal quantizer is always symmetric, i.e., if there is a quantizer boundary at y then there also is a quantizer boundary at $-y$. Thus, there is only one free parameter (i.e., quantizer boundary) in the design of quantizers with 3 and 4 quantization levels.

In Figure 4.7, we show how the rate and the relevant information behave as we vary the position of the quantizer boundaries (solid lines). The '$\times$' markers correspond to the respective MSE-optimal quantizers and the dashed lines correspond to the respective information-rate functions. We observe that the MSE-optimal quantizers almost achieve the maximum value of the relevant information (the difference is less than 1 % in all cases). More importantly, the gap to $I(R)$ is smaller (both relatively and absolutely) for the MSE-optimal quantizer than for the quantizer that maximizes the relevant information. This is because the slope of $I(R)$ is larger than the slope of the respective solid lines. The rate of the quantizers is maximized when the quantizer outputs are equally likely. This is the case at $R = \log_2(3)$ and $R = 2$, respectively. We note that the MSE-optimal quantizers outperform the corresponding maximum output entropy quantizers (cf. Section 5.1 and [73]).

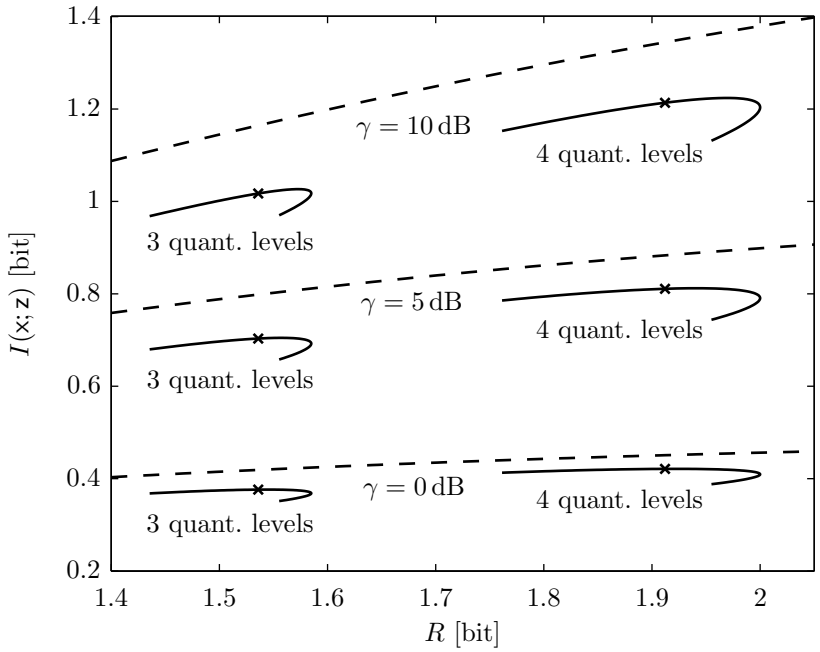


Figure 4.7: Behavior of the rate and the relevant information as the quantizer boundaries vary. MSE-optimal quantizers are indicated by '$\times$' markers and the dashed lines correspond to $I(R)$.

4.8 Discussion

In this chapter, we have used the GIB to derive closed-form expressions for the information-rate function and the rate-information function in the case of jointly Gaussian random vectors. We have shown that the optimal allocation of the compression rate corresponds to reverse waterfilling on the mode SNRs. Furthermore, MSE-optimal (noisy) source coding is suboptimal in terms of the rate-information trade-off. However, the only difference between MSE-optimal processing and rate-information-optimal processing is the rate allocation. We have proven that MSE-optimal quantization achieves the optimal rate-information trade-off when suitable linear preprocessing is used. Thereby we have also shown that the GIB is equivalent to linear filtering with subsequent MSE-optimal compression. This is important because it relates the GIB to two much more well-known concepts. Moreover, this implies that the RD theorem provides achievability and converse results for the rate-information trade-off in the Gaussian case. Finally, we have considered quantizer design and we have compared the information-rate function to the relevant information that can be preserved using finite blocklength quantizers. It turns out that it is sufficient to consider MSE-optimal quantizers. Furthermore, $I(R)$ can be closely approached as the quantization rate increases.

The results presented in this chapter can be extended to the case of complex-valued jointly Gaussian random vectors $\mathbf{x}, \mathbf{y}$ by writing the respective covariance matrices in real

form. That is, a complete statistical description of a complex Gaussian random vector $\boldsymbol{\zeta} = \boldsymbol{\zeta}_R + \sqrt{-1}\boldsymbol{\zeta}_I \in \mathbb{C}^n$ (with $\boldsymbol{\zeta}_R, \boldsymbol{\zeta}_I \in \mathbb{R}^n$) is given by the mean vector and the covariance matrix of $\widetilde{\boldsymbol{\zeta}} = \left[\boldsymbol{\zeta}_R^{\mathrm{T}}\ \boldsymbol{\zeta}_I^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{2n}$. An extension of the rate-information trade-off to jointly stationary Gaussian random processes is given in [72, Section 5]. Furthermore, our results hint at a relation between the Wiener filter and the GIB which is explored in [72, Section 3].

It is important to note that throughout this chapter, we have optimized the quantizer mapping $p(\boldsymbol{z}|\boldsymbol{y})$ for a fixed distribution $p(\boldsymbol{x})$ of the relevance variable x. In cases where $p(\boldsymbol{x})$ can be changed, it may be more interesting to consider the following joint optimization of $p(\boldsymbol{z}|\boldsymbol{y})$ and $p(\boldsymbol{x})$ with the Markov chain $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z}$:

$$\max_{\{p(\boldsymbol{z}|\boldsymbol{y}),\, p(\boldsymbol{x})\}} I(\mathsf{x};\mathsf{z}) \qquad \text{subject to} \quad I(\mathsf{y};\mathsf{z}) \leq R \quad \text{and} \quad \mathbb{E}\{\|\mathsf{x}\|_2^2\} \leq P. \tag{4.72}$$

In a communications context, (4.72) corresponds to the joint optimization of the input distribution and the channel output quantizer. Hence, the solution of (4.72) is the capacity of the quantized channel subject to a constraint on the quantization rate. The problem in (4.72) is unsolved and a Gaussian $p(\boldsymbol{x})$ is not optimal in general. To see this, consider $\mathsf{y} = \mathsf{x} + \mathsf{w}$ with $\mathsf{w} \sim \mathcal{N}(0,1)$ independent of x and $\mathsf{z} = \operatorname{sign}(\mathsf{y})$. The blue line ('$\triangledown$' marker) in Figure 4.8 shows $I(\mathsf{x};\mathsf{z})$ for an equally likely binary input $\mathsf{x} \in \{-\sqrt{\gamma}, \sqrt{\gamma}\}$. Similarly, the red line ('$\triangle$' marker) shows $I(\mathsf{x};\mathsf{z})$ for $\mathsf{x} \sim \mathcal{N}(0,\gamma)$. In both cases we have $R = H(\mathsf{z}) = 1\,\text{bit}$. We observe that the binary input significantly outperforms the Gaussian input at high SNR. However, neither of the two input distributions is optimal for all SNRs. This shows that Gaussian input distributions do not achieve the capacity of a quantized Gaussian channel for all SNRs. The joint optimization in (4.72) is hard because it couples $p(\boldsymbol{z}|\boldsymbol{y})$ and $p(\boldsymbol{x})$ in very intricate way. In particular, an alternating optimization of $p(\boldsymbol{z}|\boldsymbol{y})$ and $p(\boldsymbol{x})$ need not converge. For the special case of binary-input discrete memoryless channels, some progress towards solving (4.72) has been made. In this case, the algorithm proposed in [56] either solves (4.72) or declares an error. Unfortunately, this algorithm seems to work well only for rather small alphabet sizes of the channel output and the quantizer output.

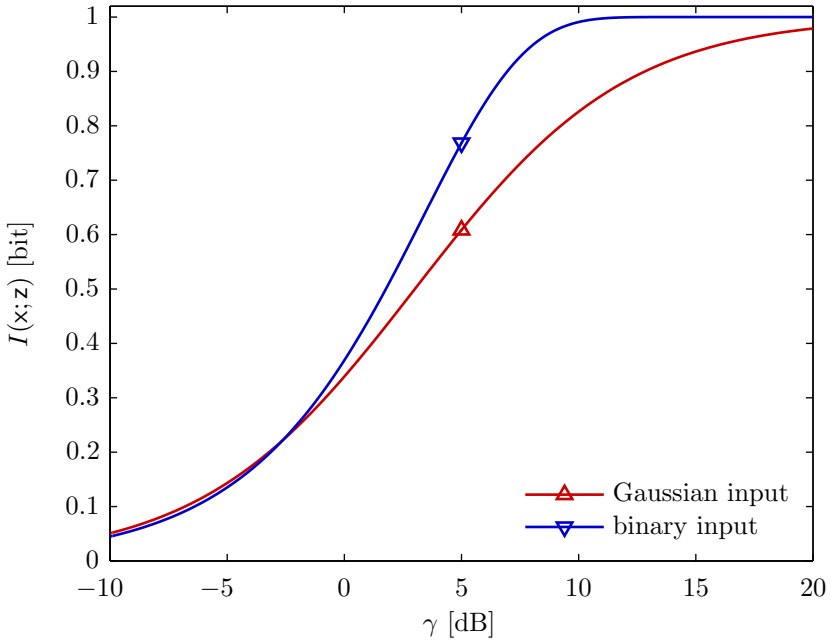


Figure 4.8: Comparison of the relevant information for 1 bit quantization of the output of a Gaussian channel with binary input and Gaussian input.

5

Quantizer Design for Communication Problems

In this chapter, we consider the design of mutual-information-optimal quantizers. We introduce the problem setup and discuss the differences to distortion-based quantizer design in Section 5.1. Next, we discuss optimal quantization in terms of the mean-square error (MSE) and we review the Lloyd-Max algorithm [66, 71] in Section 5.2. In Section 5.3, we conceive an alternating optimization algorithm for the design of scalar quantizers. This algorithm is strongly reminiscent of the famous Lloyd-Max algorithm but maximizes mutual information instead of minimizing the MSE. In Section 5.4, we present a greedy algorithm for the design of mutual-information-optimal scalar quantizers. In Section 5.5, we propose an algorithm for channel-optimized vector quantization (COVQ) which is based on the information bottleneck (IB) method and includes the design of scalar quantizers and vector quantizers as special cases. A comparison of the proposed algorithms and numerical application examples are given in Section 5.6. The discussion in Section 5.7 concludes this chapter.

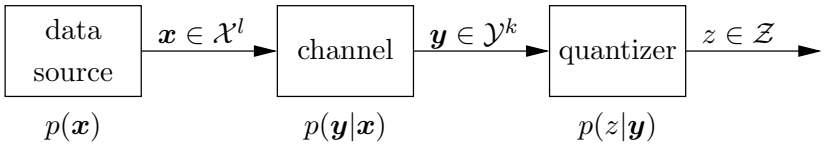


Figure 5.1: System model for quantizer design. The quantizer is designed to maximize the mutual information $I(\mathsf{x};\mathsf{z})$.

5.1 Introduction and Background

In this chapter, we devise novel algorithms for the design of low-rate quantizers for communication problems. Low-rate quantization is of interest due to constraints regarding bandwidth, memory size, power consumption, and chip area. Application examples for quantizers in digital communication systems include log-likelihood ratio (LLR) quantization for iterative decoders, quantization for receiver front-ends, and quantization in distributed systems like relay networks.

Quantization is well studied in the lossy source coding setting, and rate-distortion theory provides the corresponding fundamental performance limits. However, it is important to note that a source coding perspective is not appropriate for quantization in a communications context. Instead of representing a signal with small distortion, we are interested in maximizing the achievable rate. Hence, our objective in quantizer design is to maximize the mutual information between the data and the quantizer output.

More specifically, we consider k-dimensional vector quantization (VQ) in the setting depicted in Figure 5.1. The length-l data block $\boldsymbol{x} \in \mathcal{X}^l$ is transmitted over a channel with transition probability density function (pdf) $p(\boldsymbol{y}|\boldsymbol{x})$, yielding the length-$k$ channel output $\boldsymbol{y} \in \mathcal{Y}^k$. The quantizer $q\colon \mathcal{Y}^k \to \mathcal{Z}$ maps $\boldsymbol{y}$ to the quantizer output $z = q(\boldsymbol{y})$. We denote the number of quantization levels by $n = |\mathcal{Z}|$. The rate of the quantizer (in bits per sample) equals $\log_2(n)/k$.

Throughout this chapter we assume that $|\mathcal{X}^l| < \infty$. Additionally, we assume that the channel $p(\boldsymbol{y}|\boldsymbol{x})$ is memoryless and that the outputs of the data source are independent and identically distributed. Furthermore, the term "channel" is to be understood in a very general sense. The channel $p(\boldsymbol{y}|\boldsymbol{x})$ could for example be comprised of a modulator, a waveform channel, and a demodulator. We also allow for $l \neq k$ which is the case, e.g., when $\boldsymbol{x}$ is a vector of symbols from a higher-order signal constellation and the channel output $\boldsymbol{y}$ is the vector of LLRs for the bits corresponding to the symbols $\boldsymbol{x}$.

The quantizer $q(\cdot)$ is the solution of the following optimization problem (here, $p(\boldsymbol{x})$ and $p(\boldsymbol{y}|\boldsymbol{x})$ are fixed and known):

$$p^\star(z|\boldsymbol{y}) = \underset{p(z|\boldsymbol{y})}{\arg\max}\, I(\mathsf{x};\mathsf{z}) \qquad \text{subject to} \quad |\mathcal{Z}| = n. \tag{5.1}$$

In (5.1), the quantizer is described by the probabilistic mapping $p^\star(z|\boldsymbol{y})$. The concatenation

of the channel $p(\boldsymbol{y}|\boldsymbol{x})$ and the quantizer $q(\cdot)$ yields an overall channel with transition pdf

$$p(z|\boldsymbol{x}) = \int_{\mathcal{Y}^k} p^\star(z|\boldsymbol{y}) p(\boldsymbol{y}|\boldsymbol{x}) \mathrm{d}\boldsymbol{y}. \tag{5.2}$$

In (5.2), we have used the fact that $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z}$ is a Markov chain. The maximization of the mutual information $I(\mathsf{x};\mathsf{z})$ in (5.1) thus corresponds to maximizing the achievable rate for data transmission over the channel $p(z|\boldsymbol{x})$.

We emphasize that the quantizer design in (5.1) is substantially different from distortion-based quantizer design. The main differences are: (a) the optimization problem (5.1) involves a third random variable in addition to the quantizer input and the quantizer output, (b) the reproducer values are immaterial since mutual information depends only on the probability distributions, and (c) the problem in (5.1) is a convex *maximization* problem.

Due to (b) it suffices to choose $\mathcal{Z} = \{1, \ldots, n\}$. To see that (5.1) is a convex maximization problem, recall that $I(\mathsf{x};\mathsf{z})$ is convex in $p(z|\boldsymbol{x})$ for fixed $p(\boldsymbol{x})$. If additionally $p(\boldsymbol{y}|\boldsymbol{x})$ is fixed, then $I(\mathsf{x};\mathsf{z})$ is also convex in $p(z|\boldsymbol{y})$ due to (5.2). Furthermore, the set of valid (i.e., nonnegative and normalized) probability distributions $p(z|\boldsymbol{y})$, is a $(n-1)$-dimensional probability simplex and thus convex. Hence, (5.1) is indeed a convex maximization problem. We next show that the solution of (5.1) is a *deterministic* quantizer, i.e., we have $p^\star(z|\boldsymbol{y}) \in \{0, 1\}$. To this end, we first introduce the notion of a set that is bounded from below.

Definition 5.1. *A set $\mathcal{S} \subset \mathbb{R}^m$ is* bounded from below *if there exists an $\boldsymbol{a} \in \mathbb{R}^m$ such that $\boldsymbol{b} \succeq \boldsymbol{a}$ for all $\boldsymbol{b} \in \mathcal{S}$.*

For a feasible set $\mathcal{S}$ that is closed, convex, and bounded from below, the following proposition relates the solution of a convex maximization problem to the extreme points of its feasible set.

Proposition 5.2 (cf. [10, Proposition B.19])**.** *Let $\mathcal{S}$ be a closed convex set which is bounded from below and let $f\colon \mathcal{S} \to \mathbb{R}$ be a convex function. If f attains a maximum over $\mathcal{S}$, then it attains a maximum at some extreme point of $\mathcal{S}$.*

Note that a set which is bounded from below cannot contain a line and, hence, contains at least one extreme point. Proposition 5.2 applies to (5.1) and thus its solution is an extreme point of the probability simplex $\mathcal{P}$, i.e., we have $p^\star(z|\boldsymbol{y}) \in \{0, 1\}$ which corresponds to a deterministic quantizer.

We next briefly mention quantization for maximum output entropy (MOE). In this case, the quantizer is the solution of the following optimization problem:

$$\arg\max_{p(z|\boldsymbol{y})} H(\mathsf{z}) \qquad \text{subject to} \quad |\mathcal{Z}| = n. \tag{5.3}$$

The optimal quantizer which solves (5.3) is such that $p(z) = 1/n$, $z \in \mathcal{Z}$, and is referred to as MOE quantizer [73]. We note that the maximization of $H(\mathsf{z})$ is equivalent to the

maximization of $I(\mathsf{y};\mathsf{z}) = H(\mathsf{z}) - H(\mathsf{z}|\mathsf{y})$, i.e., the MOE quantizer maximizes the mutual information between the input and output of the quantizer. The problem (5.1) is equivalent to (5.3) if the channel $p(\boldsymbol{y}|\boldsymbol{x})$ is a one-to-one function (in this case we have $I(\mathsf{x};\mathsf{z}) = I(\mathsf{y};\mathsf{z})$). We shall use the MOE quantizer as initialization for the iterative quantizer design algorithms which we discuss in the sequel.

Quantizer design for communication problems has recently attracted some attention. In [117], LLR quantizers maximizing mutual information are designed using the iterative IB algorithm. This approach has been extended in [120] to channel output quantization for intersymbol interference channels. Maximum mutual information LLR vector quantizer design based on training data has been proposed in [22]. Quantization of conditionally Gaussian LLRs for maximum mutual information has been studied in [85]. LLR quantization for bit-interleaved coded modulation systems with soft-output demodulators has been considered in [78, 88]. In [55], LLR quantization for binary-input discrete memoryless channels (DMCs) has been studied.

5.2 MSE-Optimal Quantization and the Lloyd-Max Algorithm

In this section, we consider MSE-optimal quantization and we review the Lloyd-Max algorithm for designing an MSE-optimal scalar quantizer. We shall later see that the algorithm proposed in Section 5.3 operates in a manner that is similar to the Lloyd-Max algorithm. Using the notation introduced in the previous section, we can express the MSE distortion associated to the quantizer $q\colon \mathcal{Y}^k \to \mathcal{Z}$ as follows:

$$D = \mathbb{E}\{\|\mathbf{y} - q(\mathbf{y})\|_2^2\} = \int_{\mathcal{Y}^k} \|\boldsymbol{y} - q(\boldsymbol{y})\|_2^2 \, p(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}. \tag{5.4}$$

If we let $n = |\mathcal{Z}|$ be the number of quantization levels, then we can further rewrite (5.4) as

$$D = \sum_{i=1}^{n} \int_{\mathcal{Y}_i^k} \|\boldsymbol{y} - \boldsymbol{z}_i\|_2^2 \, p(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}, \tag{5.5}$$

where $\mathcal{Y}_i^k$ and $\boldsymbol{z}_i$, $i = 1, \ldots, n$, are the *quantization regions* and the *reproducer values*, respectively[1]. We note that $\mathcal{Y}_i^k \cap \mathcal{Y}_j^k = \emptyset$ if $i \neq j$ and $\bigcup_{i=1}^n \mathcal{Y}_i^k = \mathcal{Y}^k$. In (5.5), the quantizer is equivalently specified by $\{\mathcal{Y}_i^k\}_{i=1}^n$ and $\{\boldsymbol{z}_i\}_{i=1}^n$. An MSE-optimal quantizer minimizes D, i.e., it solves the optimization problem

$$\min_{\{\mathcal{Y}_i^k\}_{i=1}^n, \{\boldsymbol{z}_i\}_{i=1}^n} \sum_{i=1}^{n} \int_{\mathcal{Y}_i^k} \|\boldsymbol{y} - \boldsymbol{z}_i\|_2^2 \, p(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}. \tag{5.6}$$

[1]The quantization regions are sometimes referred to as boundary points, decision points, decision levels, or endpoints. Similarly, the reproducer values are also referred to as output levels, output points, or reproduction values [30, Section 5.1].

It can be shown that an MSE-optimal quantizer is always deterministic. In contrast to mutual-information-optimal quantization, the reproducer values affect the objective function in addition to the quantization regions. Therefore, $\mathcal{Z}$ is a set of n vectors of dimension k and these vectors have to be chosen optimally according to (5.6). Furthermore, the quantization regions of an MSE-optimal quantizer form a Voronoi tessellation of $\mathcal{Y}^k$ with convex Voronoi cells, i.e., the sets $\mathcal{Y}_i^k$, $i = 1, \ldots, n$, are convex sets. This is intuitive since for any given set of reproducer values $\{\boldsymbol{z}_i\}_{i=1}^n$, the following partition of $\mathcal{Y}^k$ has smallest MSE:

$$\mathcal{Y}_i^k = \{\boldsymbol{y} \in \mathcal{Y}^k \,|\, \|\boldsymbol{y} - \boldsymbol{z}_i\|_2^2 \leq \|\boldsymbol{y} - \boldsymbol{z}_j\|_2^2, j \neq i\}, \quad i = 1, \ldots, n. \tag{5.7}$$

Assuming that $\mathcal{Y}^k$ is a convex set, it is not hard to see that the quantization regions are convex as well. A tie braking strategy has to be used in (5.7) in case any $\boldsymbol{y} \in \mathcal{Y}^k$ is equidistant to two or more reproducer values. We note that the quantization regions of mutual-information-optimal quantizers need not be convex or even connected (see, e.g., [120]).

There exist several algorithms for the design of MSE-optimal quantizers, most notably the Lloyd-Max [66, 71] algorithm and the LBG algorithm [65]. The Lloyd-Max algorithm finds an MSE-optimal scalar quantizer and the LBG algorithm is an extension to vector quantizer design. Both algorithms can be used when the distribution $p(\boldsymbol{y})$ is known or unknown. In the latter case, quantizer design is performed using samples (training data) that are distributed according to $p(\boldsymbol{y})$. The Lloyd-Max algorithm and the LBG algorithm find a locally optimal solution of (5.6). In the case of discrete sources, i.e., when $p(\boldsymbol{y})$ is a probability mass function (pmf), dynamic programming can be used to find a (globally) MSE-optimal quantizer. We next describe the Lloyd-Max algorithm in more detail.

We consider MSE-optimal scalar quantization ($k = 1$) with known pdf $p(y)$ and we assume $\mathcal{Y} \subseteq \mathbb{R}$. In this case, the quantization regions are intervals on the real line. Hence, we can rewrite (5.5) as

$$D = \sum_{i=1}^{n} \int_{g_{i-1}}^{g_i} (y - z_i)^2 p(y) \mathrm{d}y, \tag{5.8}$$

where we set $g_0 = -\infty$ and $g_n = \infty$. Therefore, to find an MSE-optimal quantizer, we need to find n reproducer values $z_1, \ldots, z_n$ and $n-1$ quantizer boundaries $g_1, \ldots, g_{n-1}$ that minimize (5.8). We thus seek a solution of the following optimization problem:

$$\min_{\{g_i\}_{i=1}^{n-1}, \{z_i\}_{i=1}^{n}} \int_{-\infty}^{g_1} (y - z_i)^2 p(y) \mathrm{d}y + \sum_{i=2}^{n-1} \int_{g_{i-1}}^{g_i} (y - z_i)^2 p(y) \mathrm{d}y + \int_{g_{n-1}}^{\infty} (y - z_i)^2 p(y) \mathrm{d}y. \tag{5.9}$$

Note that the quantization regions are given as $\mathcal{Y}_i = [g_{i-1}, g_i)$, $i = 1, \ldots, n$.

We cannot minimize the MSE-distortion D directly since the problem in (5.9) is generally nonconvex. However, an alternating minimization can be used to obtain a locally optimal

solution of (5.9). A necessary condition for the optimality of the quantizer boundaries is

$$\frac{\partial D}{\partial g_j} = (g_j - z_j)^2 p_{\mathsf{y}}(g_j) - (g_j - z_{j+1})^2 p_{\mathsf{y}}(g_j) = 0, \qquad j = 1, \ldots, n-1. \tag{5.10}$$

Assuming $p_{\mathsf{y}}(g_j) > 0$ in (5.10) and solving for g_j yields

$$g_j = \frac{z_{j+1} + z_j}{2}, \qquad j = 1, \ldots, n-1. \tag{5.11}$$

Thus, for fixed reproducer values, the optimal quantizer boundaries are given as the arithmetic mean of their neighboring reproducer values. Next, we compute the partial derivatives with respect to the reproducer values. We have

$$\frac{\partial D}{\partial z_j} = -2 \int_{g_{j-1}}^{g_j} (y - z_j) p(y) \mathrm{d}y, \qquad j = 1, \ldots, n. \tag{5.12}$$

Setting the derivatives in (5.12) to zero and solving for z_j yields

$$z_j = \frac{\displaystyle\int_{g_{j-1}}^{g_j} y p(y) \mathrm{d}y}{\displaystyle\int_{g_{j-1}}^{g_j} p(y) \mathrm{d}y}, \qquad j = 1, \ldots, n. \tag{5.13}$$

Thus, for fixed quantizer boundaries, the optimal reproducer values are the centroids of their respective quantization region.

The Lloyd-Max algorithm uses the coupled equations (5.11) and (5.13) to iteratively find a locally optimal quantizer. The algorithm is initialized with a guess for either the reproducer values or the quantizer boundaries. Next, the quantizer boundaries and the reproducer values are alternatingly updated using (5.11) and (5.13). The algorithm stops when the largest change in the reproducer values between two iterations is below a prescribed threshold or if a certain number of iterations has been performed.

In addition to the number of iterations and the stopping threshold, the result of the Lloyd-Max algorithm is affected by the initialization. We have found that using the MOE quantizer [73] as initialization yields good results. Finally, we note that the Lloyd-Max algorithm can also be used to numerically obtain optimized quantizers for distortion measures other than the squared-error distortion.

5.3 Scalar Quantizer Design for Maximum Mutual Information

We next devise an algorithm for scalar ($k = 1$) quantizer design which maximizes the mutual information $I(\mathsf{x}; \mathsf{z})$ and operates in a similar manner as the Lloyd-Max algorithm. A related approach has been proposed in [85]. However, in contrast to our work, [85] is restricted to the binary case, i.e., $|\mathcal{X}^l| = 2$, and to the quantization of conditionally Gaussian LLRs. In

what follows, we assume that y is a continuous random variable with $\mathcal{Y} \subseteq \mathbb{R}$. Furthermore, we let $\mathcal{Z} = \{1, \ldots, n\}$, where n is the number of quantization levels.

The objective function in (5.1) can be written explicitly in terms of $p(z|y)$ as follows:

$$I(\mathsf{x};\mathsf{z}) = \sum_{\boldsymbol{x}\in\mathcal{X}^l} p(\boldsymbol{x}) \sum_{z\in\mathcal{Z}} p(z|\boldsymbol{x}) \log \frac{p(z|\boldsymbol{x})}{p(z)} \tag{5.14}$$

$$= \sum_{\boldsymbol{x}\in\mathcal{X}^l} p(\boldsymbol{x}) \sum_{z\in\mathcal{Z}} \int_{\mathcal{Y}} p(z|y)p(y|\boldsymbol{x})\mathrm{d}y \log \frac{\displaystyle\int_{\mathcal{Y}} p(z|y)p(y|\boldsymbol{x})\mathrm{d}y}{\displaystyle\int_{\mathcal{Y}} p(z|y)p(y)\mathrm{d}y} \tag{5.15}$$

$$= \sum_{\boldsymbol{x}\in\mathcal{X}^l} p(\boldsymbol{x}) \sum_{z\in\mathcal{Z}} \int_{\mathcal{Y}_z} p(y|\boldsymbol{x})\mathrm{d}y \log \frac{\displaystyle\int_{\mathcal{Y}_z} p(y|\boldsymbol{x})\mathrm{d}y}{\displaystyle\int_{\mathcal{Y}_z} p(y)\mathrm{d}y}. \tag{5.16}$$

In (5.16), we have used the fact that the optimal quantizer is deterministic (i.e., $p(z|y) \in \{0, 1\}$). We can thus write the quantization regions as

$$\mathcal{Y}_z = \{y \in \mathcal{Y} \,|\, p(z|y) = 1\}, \qquad z = 1, \ldots, n. \tag{5.17}$$

Finding an optimal quantizer therefore amounts to finding the quantization regions (5.17) such that (5.16) is maximized. Hence, we can write the quantizer design problem as follows:

$$\max_{\mathcal{Y}_1,\ldots,\mathcal{Y}_n} \sum_{\boldsymbol{x}\in\mathcal{X}^l} p(\boldsymbol{x}) \sum_{z\in\mathcal{Z}} \int_{\mathcal{Y}_z} p(y|\boldsymbol{x})\mathrm{d}y \log \frac{\displaystyle\int_{\mathcal{Y}_z} p(y|\boldsymbol{x})\mathrm{d}y}{\displaystyle\int_{\mathcal{Y}_z} p(y)\mathrm{d}y}. \tag{5.18}$$

As we have mentioned in the previous section, the optimal quantization regions need not be convex sets. However, it can be shown that the quantization regions are convex if y is a posterior probability for x. This is a sufficient condition for the convexity of the quantization regions which is a consequence of [14, Theorem 1]. In the sequel, we assume that the optimal quantization regions are indeed convex sets.

5.3.1 Nonbinary Case

We first consider the nonbinary case, i.e., we have $|\mathcal{X}^l| > 2$. Assuming that the optimal quantization regions are convex sets allows us to rewrite (5.16) as follows:

$$I(\boldsymbol{g}) = I(g_1, \ldots, g_{n-1}) = \sum_{\boldsymbol{x}\in\mathcal{X}^l} p(\boldsymbol{x}) \sum_{i=1}^{n} \int_{g_{i-1}}^{g_i} p(y|\boldsymbol{x})\mathrm{d}y \log \frac{\displaystyle\int_{g_{i-1}}^{g_i} p(y|\boldsymbol{x})\mathrm{d}y}{\displaystyle\int_{g_{i-1}}^{g_i} p(y)\mathrm{d}y}. \tag{5.19}$$

Here, we use the notation $I(\boldsymbol{g})$, with $\boldsymbol{g} = (g_1 \cdots g_{n-1})^{\mathrm{T}}$, to emphasize that for fixed $p(\boldsymbol{x})$ and $p(y|\boldsymbol{x})$ the mutual information $I(\mathsf{x};\mathsf{z})$ is determined solely by the quantizer boundaries. We again set $g_0 = -\infty$ and $g_n = \infty$. The quantizer design problem (5.18) can thus be rewritten as follows:

$$\max_{g_1,\ldots,g_{n-1}} \sum_{\boldsymbol{x}\in\mathcal{X}^l} p(\boldsymbol{x}) \sum_{i=1}^{n} \int_{g_{i-1}}^{g_i} p(y|\boldsymbol{x})\mathrm{d}y \log \frac{\displaystyle\int_{g_{i-1}}^{g_i} p(\boldsymbol{x}|y)p(y)\mathrm{d}y}{p(\boldsymbol{x}) \displaystyle\int_{g_{i-1}}^{g_i} p(y)\mathrm{d}y}, \tag{5.20}$$

where we have used Bayes' rule in the numerator of the logarithm in (5.19). Next, we define

$$h_i^\star(\boldsymbol{x}) \triangleq \frac{\displaystyle\int_{g_{i-1}}^{g_i} p(\boldsymbol{x}|y)p(y)\mathrm{d}y}{\displaystyle\int_{g_{i-1}}^{g_i} p(y)\mathrm{d}y}, \qquad i = 1,\ldots,n, \tag{5.21}$$

which allows us to rewrite the objective function in (5.20) as

$$I(\boldsymbol{g}) = \sum_{\boldsymbol{x}\in\mathcal{X}^l} p(\boldsymbol{x}) \sum_{i=1}^{n} \int_{g_{i-1}}^{g_i} p(y|\boldsymbol{x})\mathrm{d}y \log \frac{h_i^\star(\boldsymbol{x})}{p(\boldsymbol{x})}. \tag{5.22}$$

We note that $h_i^\star(\boldsymbol{x})$ is the *a posteriori* probability (APP) $\mathbb{P}\{\mathsf{x}=\boldsymbol{x}|z=i\}$. Next, we let $h_i(\boldsymbol{x})$, $i = 1,\ldots,n$, be *arbitrary* probability distributions on $\mathcal{X}^l$. We define the modified objective function

$$I\big(\boldsymbol{g}, \boldsymbol{h}(\boldsymbol{x})\big) \triangleq \sum_{\boldsymbol{x}\in\mathcal{X}^l} p(\boldsymbol{x}) \sum_{i=1}^{n} \int_{g_{i-1}}^{g_i} p(y|\boldsymbol{x})\mathrm{d}y \log \frac{h_i(\boldsymbol{x})}{p(\boldsymbol{x})}, \tag{5.23}$$

where $\boldsymbol{h}(\boldsymbol{x}) = \big(h_1(\boldsymbol{x}) \cdots h_n(\boldsymbol{x})\big)^{\mathrm{T}}$. The following result relates (5.23) to (5.22).

Proposition 5.3. *The functions $I\big(\boldsymbol{g}, \boldsymbol{h}(\boldsymbol{x})\big)$ and $I(\boldsymbol{g})$ are related as*

$$\max_{\boldsymbol{h}(\boldsymbol{x})} I\big(\boldsymbol{g}, \boldsymbol{h}(\boldsymbol{x})\big) = I(\boldsymbol{g}), \tag{5.24}$$

where the maximum in (5.24) is achieved by (5.21). Therefore, the quantizer design problem can be rewritten as follows:

$$\max_{\boldsymbol{g}} I(\boldsymbol{g}) = \max_{\boldsymbol{g}} \max_{\boldsymbol{h}(\boldsymbol{x})} I\big(\boldsymbol{g}, \boldsymbol{h}(\boldsymbol{x})\big). \tag{5.25}$$

Proof: See Appendix C.1. ∎

We note that the approach of rewriting the original problem as in (5.25) is similar to the Blahut-Arimoto algorithm [3, 11]. Proposition 5.3 allows us to approach the quantizer design problem via alternating maximization. Next, we compute the partial derivatives

$\partial I\big(\boldsymbol{g}, \boldsymbol{h}(\boldsymbol{x})\big)/\partial g_j$, $j = 1, \ldots, n-1$. We have

$$\frac{\partial I\big(\boldsymbol{g}, \boldsymbol{h}(\boldsymbol{x})\big)}{\partial g_j} = \frac{\partial}{\partial g_j} \sum_{\boldsymbol{x} \in \mathcal{X}^l} p(\boldsymbol{x}) \sum_{i=1}^{n} \int_{g_{i-1}}^{g_i} p(y|\boldsymbol{x}) \mathrm{d}y \log \frac{h_i(\boldsymbol{x})}{p(\boldsymbol{x})} \tag{5.26}$$

$$= \sum_{\boldsymbol{x} \in \mathcal{X}^l} p(\boldsymbol{x}) \left[p_{\mathsf{y}|\boldsymbol{x}}(g_j|\boldsymbol{x}) \log \frac{h_j(\boldsymbol{x})}{p(\boldsymbol{x})} - p_{\mathsf{y}|\boldsymbol{x}}(g_j|\boldsymbol{x}) \log \frac{h_{j+1}(\boldsymbol{x})}{p(\boldsymbol{x})} \right] \tag{5.27}$$

$$= \sum_{\boldsymbol{x} \in \mathcal{X}^l} p(\boldsymbol{x}) p_{\mathsf{y}|\boldsymbol{x}}(g_j|\boldsymbol{x}) \log \frac{h_j(\boldsymbol{x})}{h_{j+1}(\boldsymbol{x})} \tag{5.28}$$

$$= p_{\mathsf{y}}(g_j) \sum_{\boldsymbol{x} \in \mathcal{X}^l} p_{\mathbf{x}|y}(\boldsymbol{x}|g_j) \log \frac{h_j(\boldsymbol{x})}{h_{j+1}(\boldsymbol{x})}, \qquad j = 1, \ldots, n-1. \tag{5.29}$$

Setting the above derivatives to zero and assuming $p_{\mathsf{y}}(g_j) > 0$ in (5.29) yields the following necessary conditions for optimality of the quantizer boundaries:

$$\mathbb{E}\left\{ \log \frac{h_j(\mathbf{x})}{h_{j+1}(\mathbf{x})} \,\middle|\, \mathsf{y} = g_j \right\} = 0, \qquad j = 1, \ldots, n-1. \tag{5.30}$$

Any suitable root-finding method (e.g., Brent's method [13, Chapter 4]) can be used to find the quantizer boundaries such that (5.30) is fulfilled.

Our algorithm starts with an initialization for $\boldsymbol{g}$ such that $p_{\mathsf{y}}(g_j) > 0$, $j = 1, \ldots, n-1$. The quantities $\boldsymbol{h}(\boldsymbol{x})$ and $\boldsymbol{g}$ are then updated alternatingly using (5.21) and (5.30). The algorithm stops if the increase in $I(\mathbf{x}; \mathsf{z})$ between two iterations is below a prescribed threshold or if a certain number of iterations has been performed. Algorithm 5.1 summarizes the proposed algorithm for scalar quantizer design in the nonbinary case. Although we have observed excellent convergence behavior, we were unable to formally prove the convergence of this algorithm. The choice of the initialization for $\boldsymbol{g}$ may affect the resulting quantizer. In principle any initialization with $p_{\mathsf{y}}(g_j) > 0$, $j = 1, \ldots, n-1$, is acceptable. However, we have found that initializing $\boldsymbol{g}$ using the MOE quantizer yields good results.

5.3.2 Binary Case

We next specialize our algorithm to the binary case, i.e., $\mathsf{x} \in \mathcal{X}$ is a binary random variable ($l = 1$). Without loss of generality we let $\mathcal{X} = \{-1, 1\}$. In this case, we can elegantly reformulate Algorithm 5.1 in terms of LLRs. In particular, we rewrite (5.21) as

$$h_i^\star(x) = \frac{\displaystyle\int_{g_{i-1}}^{g_i} p(y|x)p(x)\mathrm{d}y}{\displaystyle\int_{g_{i-1}}^{g_i} p(y|x)p(x)\mathrm{d}y + \int_{g_{i-1}}^{g_i} p(y|-x)p(-x)\mathrm{d}y} = \frac{1}{1 + e^{-xL_i}}, \qquad i = 1, \ldots, n. \tag{5.31}$$

Algorithm 5.1 *Scalar quantizer design for maximum mutual information (nonbinary case).*

Input: $\mathcal{X}^l$, $\mathcal{Y}$, $\mathcal{Z}$, $p(\boldsymbol{x}, y)$, $\varepsilon > 0$, $M \in \mathbb{N}$

Initialization: $\eta \leftarrow \infty$, $m \leftarrow 1$, $n \leftarrow |\mathcal{Z}|$, choose $\boldsymbol{g}^{(0)}$ such that $g_1^{(0)} < \cdots < g_{n-1}^{(0)}$ and $p_{\mathsf{y}}\big(g_j^{(0)}\big) > 0$, $j = 1, \ldots, n-1$, $g_0^{(0)} \leftarrow -\infty$, $g_n^{(0)} \leftarrow \infty$

$h_j^{(0)}(\boldsymbol{x}) \leftarrow \displaystyle\int_{g_{j-1}^{(0)}}^{g_j^{(0)}} p(\boldsymbol{x}, y)\mathrm{d}y, \qquad j = 1, \ldots, n$

$h_j^{(0)}(\boldsymbol{x}) \leftarrow h_j^{(0)}(\boldsymbol{x}) / \sum_{\boldsymbol{x} \in \mathcal{X}^l} h_j^{(0)}(\boldsymbol{x}), \qquad j = 1, \ldots, n$

$I^{(0)} \leftarrow \displaystyle\sum_{\boldsymbol{x} \in \mathcal{X}^l} p(\boldsymbol{x}) \sum_{j=1}^{n} \int_{g_{j-1}^{(0)}}^{g_j^{(0)}} p(y|\boldsymbol{x})\mathrm{d}y \log \frac{h_j^{(0)}(\boldsymbol{x})}{p(\boldsymbol{x})}$

while $\eta \geq \varepsilon$ **and** $m \leq M$ **do**

  $g_0^{(m)} \leftarrow -\infty$, $g_n^{(m)} \leftarrow \infty$

  $g_j^{(m)} \leftarrow$ root of $\mathbb{E}\left\{ \log \dfrac{h_j^{(m-1)}(\mathsf{x})}{h_{j+1}^{(m-1)}(\mathsf{x})} \,\middle|\, \mathsf{y} = y \right\}, \qquad j = 1, \ldots, n-1$

  $h_j^{(m)}(\boldsymbol{x}) \leftarrow \displaystyle\int_{g_{j-1}^{(m)}}^{g_j^{(m)}} p(\boldsymbol{x}, y)\mathrm{d}y, \qquad j = 1, \ldots, n$

  $h_j^{(m)}(\boldsymbol{x}) \leftarrow h_j^{(m)}(\boldsymbol{x}) / \sum_{\boldsymbol{x} \in \mathcal{X}^l} h_j^{(m)}(\boldsymbol{x}), \qquad j = 1, \ldots, n$

  $I^{(m)} \leftarrow \displaystyle\sum_{\boldsymbol{x} \in \mathcal{X}^l} p(\boldsymbol{x}) \sum_{j=1}^{n} \int_{g_{j-1}^{(m)}}^{g_j^{(m)}} p(y|\boldsymbol{x})\mathrm{d}y \log \frac{h_j^{(m)}(\boldsymbol{x})}{p(\boldsymbol{x})}$

  $\eta \leftarrow \big(I^{(m)} - I^{(m-1)}\big) / I^{(m)}$

  $m \leftarrow m + 1$

end while

Output: quantizer boundaries $\boldsymbol{g}^{(m-1)}$ and APPs $\boldsymbol{h}^{(m-1)}(\boldsymbol{x})$

The LLR L_i in (5.31) equals

$$L_i = \log \frac{\mathbb{P}\{\mathsf{x}=1|\mathsf{z}=i\}}{\mathbb{P}\{\mathsf{x}=-1|\mathsf{z}=i\}} = \log \frac{\mathbb{P}\{\mathsf{x}=1,\mathsf{z}=i\}}{\mathbb{P}\{\mathsf{x}=-1,\mathsf{z}=i\}} = \log \frac{\mathbb{P}\{\mathsf{x}=1\} \int_{g_{i-1}}^{g_i} p(y|\mathsf{x}=1)\mathrm{d}y}{\mathbb{P}\{\mathsf{x}=-1\} \int_{g_{i-1}}^{g_i} p(y|\mathsf{x}=-1)\mathrm{d}y}. \quad (5.32)$$

Hence, L_i is the posterior LLR for x when $\mathsf{z}=i$, i.e., we can identify each quantizer output with its corresponding LLR L_z, $z \in \mathcal{Z} = \{1,\ldots,n\}$. Using (5.31), we can rewrite the necessary optimality condition (5.30) for the quantizer boundaries as follows:

$$\mathbb{P}\{\mathsf{x}=1|\mathsf{y}=g_j\} \log \frac{1+e^{-L_{j+1}}}{1+e^{-L_j}} + \mathbb{P}\{\mathsf{x}=-1|\mathsf{y}=g_j\} \log \frac{1+e^{L_{j+1}}}{1+e^{L_j}} = 0, \qquad j=1,\ldots,n-1. \quad (5.33)$$

Next, we further rewrite (5.33) in terms of the LLR $L_{\mathsf{x}}(g_j) = \log \mathbb{P}\{\mathsf{x}=1|\mathsf{y}=g_j\} - \log \mathbb{P}\{\mathsf{x}=-1|\mathsf{y}=g_j\}$. We have

$$L_{\mathsf{x}}(g_j) = \log \frac{\log \frac{1+e^{L_{j+1}}}{1+e^{L_j}}}{\log \frac{1+e^{-L_j}}{1+e^{-L_{j+1}}}}, \qquad j=1,\ldots,n-1. \quad (5.34)$$

A suitable root-finding method can again be used to find the quantizer boundaries such that (5.34) is fulfilled. We note that (5.34) has a unique solution if $L_{\mathsf{x}}(y)$ is strictly increasing in y (cf. Appendix C.2). In the special case of LLR quantization, y is an LLR and we thus have $L_{\mathsf{x}}(g_j) = g_j$ (cf. Lemma 3.2). This yields the following closed-form solution for the quantizer boundaries:

$$g_j = \log \frac{\log \frac{1+e^{L_{j+1}}}{1+e^{L_j}}}{\log \frac{1+e^{-L_j}}{1+e^{-L_{j+1}}}}, \qquad j=1,\ldots,n-1. \quad (5.35)$$

In the binary case, our algorithm starts with an initialization for $\boldsymbol{g}$ such that $p_{\mathsf{y}}(g_j) > 0$, $j=1,\ldots,n-1$. The quantities $\{L_i\}_{i=1}^n$ and $\boldsymbol{g}$ are then updated alternatingly using (5.32) and (5.34). The algorithm stops if the increase in $I(\mathsf{x};\mathsf{z})$ between two iterations is below a prescribed threshold or if a certain number of iterations has been performed. Algorithm 5.2 summarizes the proposed algorithm for scalar quantizer design in the binary case. The following proposition gives conditions which imply convergence of the proposed algorithm to a locally optimal quantizer.

Proposition 5.4. *In the binary case, the proposed algorithm converges to a locally optimal solution of (5.20) if the LLR $L_{\mathsf{x}}(y)$ is strictly increasing in y. For the special case of LLR quantization, the proposed algorithm thus always finds a locally optimal quantizer.*

Proof: See Appendix C.2. ■

Algorithm 5.2 *Scalar quantizer design for maximum mutual information (binary case).*

Input: $\mathcal{Y}$, $\mathcal{Z}$, $p(x,y)$, $\varepsilon > 0$, $M \in \mathbb{N}$

Initialization: $\mathcal{X} = \{-1, 1\}$, $\eta \leftarrow \infty$, $m \leftarrow 1$, $n \leftarrow |\mathcal{Z}|$, choose $\boldsymbol{g}^{(0)}$ such that $g_1^{(0)} < \cdots < g_{n-1}^{(0)}$ and $p_{\mathsf{y}}\big(g_j^{(0)}\big) > 0$, $j = 1, \ldots, n-1$, $g_0^{(0)} \leftarrow -\infty$, $g_n^{(0)} \leftarrow \infty$

$$L_j^{(0)} \leftarrow \log \int_{g_{j-1}^{(0)}}^{g_j^{(0)}} p(y|\mathsf{x}{=}1)\mathbb{P}\{\mathsf{x}{=}1\}\mathrm{d}y - \log \int_{g_{j-1}^{(0)}}^{g_j^{(0)}} p(y|\mathsf{x}{=}{-1})\mathbb{P}\{\mathsf{x}{=}{-1}\}\mathrm{d}y, \qquad j = 1, \ldots, n$$

$$I^{(0)} \leftarrow \sum_{x \in \mathcal{X}} p(x) \sum_{j=1}^{n} \int_{g_{j-1}^{(0)}}^{g_j^{(0)}} p(y|x)\mathrm{d}y \log \frac{1}{p(x)\Big(1 + e^{-xL_j^{(0)}}\Big)}$$

while $\eta \geq \varepsilon$ **and** $m \leq M$ **do**

$$g_0^{(m)} \leftarrow -\infty,\ g_n^{(m)} \leftarrow \infty$$

$$g_j^{(m)} \leftarrow \text{root of } L_{\mathsf{x}}(y) + \log\log \frac{1 + e^{-L_j^{(m-1)}}}{1 + e^{-L_{j+1}^{(m-1)}}} - \log\log \frac{1 + e^{L_{j+1}^{(m-1)}}}{1 + e^{L_j^{(m-1)}}}, \qquad j = 1, \ldots, n-1$$

$$L_j^{(m)} \leftarrow \log \int_{g_{j-1}^{(m)}}^{g_j^{(m)}} p(y|\mathsf{x}{=}1)\mathbb{P}\{\mathsf{x}{=}1\}\mathrm{d}y - \log \int_{g_{j-1}^{(m)}}^{g_j^{(m)}} p(y|\mathsf{x}{=}{-1})\mathbb{P}\{\mathsf{x}{=}{-1}\}\mathrm{d}y, \qquad j = 1, \ldots, n$$

$$I^{(m)} \leftarrow \sum_{x \in \mathcal{X}} p(x) \sum_{j=1}^{n} \int_{g_{j-1}^{(m)}}^{g_j^{(m)}} p(y|x)\mathrm{d}y \log \frac{1}{p(x)\Big(1 + e^{-xL_j^{(m)}}\Big)}$$

$$\eta \leftarrow \big(I^{(m)} - I^{(m-1)}\big)/I^{(m)}$$

$$m \leftarrow m + 1$$

end while

Output: quantizer boundaries $\boldsymbol{g}^{(m-1)}$ and posterior LLRs $L_j^{(m-1)}$, $j = 1, \ldots, n$

5.4 A Greedy Algorithm for Scalar Quantizer Design

In this section, we again consider scalar quantizer design and we assume that the optimal quantization regions are convex sets. Instead of the alternating optimization approach of Section 5.3, the algorithm proposed in this section directly optimizes the quantizer boundaries $g_1, \ldots, g_{n-1}$. Maximizing the mutual information $I(\mathsf{x};\mathsf{z}) = H(\mathsf{x}) - H(\mathsf{x}|\mathsf{z})$ is equivalent to minimizing the conditional entropy $H(\mathsf{x}|\mathsf{z})$. Hence, we want to minimize the following objective function (here, $g_0 = -\infty$, $g_n = \infty$, and n is the number of quantization levels):

$$H(\boldsymbol{g}) = H(g_1, \ldots, g_{n-1}) = \sum_{\boldsymbol{x} \in \mathcal{X}^l} p(\boldsymbol{x}) \sum_{i=1}^{n} \int_{g_{i-1}}^{g_i} p(y|\boldsymbol{x}) \mathrm{d}y \log \int_{g_{i-1}}^{g_i} p(y|\boldsymbol{x}) \mathrm{d}y, \tag{5.36}$$

where we use $H(\boldsymbol{g})$ to denote the dependence of $H(\mathsf{x}|\mathsf{z})$ on the quantizer boundaries. Minimizing (5.36) with respect to $\boldsymbol{g} = (g_1 \cdots g_{n-1})^{\mathrm{T}}$ is difficult since $H(\boldsymbol{g})$ is not convex in $\boldsymbol{g}$. We thus propose to iteratively optimize one quantizer boundary at a time in a greedy fashion. This approach allows us to find a locally optimal quantizer up to a desired accuracy.

The proposed algorithm starts with an initialization $\boldsymbol{g}$ for the quantizer boundaries. Next, a set of candidate quantizer boundaries is generated based on $\boldsymbol{g}$. These candidates, denoted by $\widetilde{\boldsymbol{g}}_j$, $j = 1, \ldots, 2(n-1)$, are given as

$$\widetilde{\boldsymbol{g}}_{2j-1} = (g_1 \cdots g_{j-1} \ \widetilde{g}_j^- \ g_{j+1} \cdots g_{n-1})^{\mathrm{T}}, \qquad j = 1, \ldots, n-1, \tag{5.37a}$$

$$\widetilde{\boldsymbol{g}}_{2j} = (g_1 \cdots g_{j-1} \ \widetilde{g}_j^+ \ g_{j+1} \cdots g_{n-1})^{\mathrm{T}}, \qquad j = 1, \ldots, n-1, \tag{5.37b}$$

where $\widetilde{g}_j^- < g_j < \widetilde{g}_j^+$. Of course, the modified quantizer boundaries must be such that the elements of the vector $\widetilde{\boldsymbol{g}}_j$ are still sorted, i.e., we have $g_{j-1} < \widetilde{g}_j^- < g_{j+1}$ and $g_{j-1} < \widetilde{g}_j^+ < g_{j+1}$. Next, (5.36) is evaluated for all candidates and the difference to $H(\boldsymbol{g})$ is computed. We have

$$\Delta_j = H(\boldsymbol{g}) - H(\widetilde{\boldsymbol{g}}_j), \qquad j = 1, \ldots, 2(n-1), \tag{5.38}$$

and we let $i = \arg\max_j \Delta_j$ be the index of the candidate that yields the largest difference in the objective function value. If $\Delta_i \leq 0$, then none of the candidates improves over $\boldsymbol{g}$. In this case, the generation of the candidates is either refined or the algorithm is terminated. Otherwise, we have $\Delta_i > 0$ and thus among all candidates $\widetilde{\boldsymbol{g}}_i$ yields the largest improvement. Since the algorithm operates in greedy manner, we use $\widetilde{\boldsymbol{g}}_i$ as updated quantizer boundaries. The next iteration starts with the generation of new candidates based on $\widetilde{\boldsymbol{g}}_i$. The algorithm terminates after a certain number of iterations and refinements of the candidate generation.

We next describe the generation of the candidate quantizer boundaries. For the boundaries g_j, $j = 2, \ldots, n-2$, we generate the candidates as follows:

$$\widetilde{g}_j^- = g_j - \frac{g_j - g_{j-1}}{2^r}, \qquad j = 2, \ldots, n-2, \tag{5.39a}$$

$$\widetilde{g}_j^+ = g_j + \frac{g_{j+1} - g_j}{2^r}, \qquad j = 2, \ldots, n-2, \tag{5.39b}$$

where $r \geq 1$ is the refinement level. For the boundaries g_1 and g_{n-1} we propose to use the candidates

$$\widetilde{g}_1^- = g_1 - \frac{g_2 - g_1}{2^r}, \qquad \widetilde{g}_{n-1}^- = g_{n-1} - \frac{g_{n-1} - g_{n-2}}{2^r}, \tag{5.40a}$$

$$\widetilde{g}_1^+ = g_1 + \frac{g_2 - g_1}{2^r}, \qquad \widetilde{g}_{n-1}^+ = g_{n-1} + \frac{g_{n-1} - g_{n-2}}{2^r}. \tag{5.40b}$$

We start the algorithm with $r = 1$ and we increment r by 1 to refine the candidate generation. Algorithm 5.3 summarizes the proposed greedy algorithm for scalar quantizer design. The effectiveness and the convergence behavior of this algorithm is demonstrated in Section 5.6. We note that the proposed algorithm converges to a locally optimum quantizer as we increase r. The evaluation of $H(\boldsymbol{g})$ for all candidates can be carried out in an efficient manner since for each candidate only two out of n integrals in (5.36) change.

We note that the proposed greedy algorithm is attractive due to its conceptual and computational simplicity. In particular, root-finding is avoided which is in contrast to the alternating optimization algorithms of Section 5.3. Furthermore, a suitably modified version of our greedy algorithm can be used to find a locally optimal quantizer when y is a discrete random variable, i.e., when $|\mathcal{Y}| < \infty$.

Algorithm 5.3 *Greedy algorithm for scalar quantizer design.*

Input: $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, $p(\boldsymbol{x}, y)$, $M, R \in \mathbb{N}$

Initialization: $m \leftarrow 1$, $r \leftarrow 1$, $n \leftarrow |\mathcal{Z}|$, choose $\boldsymbol{g}^{(0)}$ such that $g_1^{(0)} < \cdots < g_{n-1}^{(0)}$

while $m \leq M$ **and** $r \leq R$ **do**
  $\widetilde{\boldsymbol{g}}_j \leftarrow$ candidates based on $\boldsymbol{g}^{(m-1)}$ and r, cf. (5.37), (5.39), (5.40), $j = 1, \ldots, 2(n-1)$
  $\Delta_j \leftarrow H(\boldsymbol{g}^{(m-1)}) - H(\widetilde{\boldsymbol{g}}_j), \qquad j = 1, \ldots, 2(n-1)$
  $i \leftarrow \underset{j \in \{1,\ldots,2(n-1)\}}{\arg\max} \Delta_j$
  if $\Delta_i > 0$ **then**
    $\boldsymbol{g}^{(m)} \leftarrow \widetilde{\boldsymbol{g}}_i$
  else
    $r \leftarrow r + 1$
    $\boldsymbol{g}^{(m)} \leftarrow \boldsymbol{g}^{(m-1)}$
  end if
  $m \leftarrow m + 1$
end while

Output: quantizer boundaries $\boldsymbol{g}^{(m-1)}$

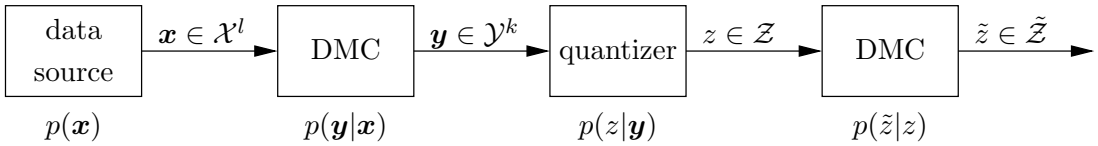


Figure 5.2: System model for COVQ. The quantizer is designed to maximize the mutual information $I(\mathsf{x};\tilde{\mathsf{z}})$.

5.5 Channel-Optimized Vector Quantization for Maximum Mutual Information

We next extend the model depicted in Figure 5.1 to channel-optimized quantization (cf. Figure 5.2). In this model, the quantizer output is additionally transmitted over a DMC with transition pmf $p(\tilde{z}|z)$. The quantizer is designed to maximize the mutual information $I(\mathsf{x};\tilde{\mathsf{z}})$, i.e., its design incorporates the channel $p(\tilde{z}|z)$ and, hence, the quantizer is called *channel-optimized*. Example scenarios in which quantizer outputs are corrupted by a subsequent channel are distributed systems like relay networks with noisy links (cf. Chapter 6), distributed inference schemes in sensor networks, and practical receiver implementations with unreliable memories [80, 89].

In what follows, we conceive an algorithm for the design of channel-optimized vector quantizers that maximize $I(\mathsf{x};\tilde{\mathsf{z}})$. COVQ is well-known in the lossy joint source-channel coding setting [25, 26]. However, we appear to be the first to address and solve the problem of designing channel-optimized vector quantizers for maximum mutual information. In the sequel, we shall refer to $p(\boldsymbol{y}|\boldsymbol{x})$ and $p(\tilde{z}|z)$ as the "unquantized channel" and the "forward channel", respectively. Furthermore, we assume that these channels are DMCs and hence the sets $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, $\tilde{\mathcal{Z}}$ are of finite cardinality. Since VQ is a special case of COVQ, the proposed algorithm can also be used for the design of conventional (non-channel-optimized) vector quantizers maximizing the mutual information $I(\mathsf{x};\mathsf{z})$.

Our aim is to find a quantizer that maximizes the achievable rate over the end-to-end channel $p(\tilde{z}|\boldsymbol{x})$. Hence, the channel-optimized vector quantizer with n quantization levels is given as

$$p^\star(z|\boldsymbol{y}) = \underset{p(z|\boldsymbol{y})}{\arg\max}\, I(\mathsf{x};\tilde{\mathsf{z}}) \qquad \text{subject to} \quad |\mathcal{Z}| = n. \tag{5.41}$$

Since $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z} \leftrightarrow \tilde{\mathsf{z}}$ forms a Markov chain, we have

$$p(\tilde{z}|\boldsymbol{x}) = \sum_{z\in\mathcal{Z}} p(\tilde{z}|z) \sum_{\boldsymbol{y}\in\mathcal{Y}^k} p(z|\boldsymbol{y})p(\boldsymbol{y}|\boldsymbol{x}). \tag{5.42}$$

Expanding the mutual information $I(\mathsf{x};\tilde{\mathsf{z}})$ using (5.42) allows us to rewrite (5.41) as follows:

$$p^\star(z|\boldsymbol{y}) = \underset{p(z|\boldsymbol{y})}{\arg\max} \sum_{\boldsymbol{x}\in\mathcal{X}^l} p(\boldsymbol{x}) \sum_{\tilde{z}\in\tilde{\mathcal{Z}}} \sum_{z\in\mathcal{Z}} p(\tilde{z}|z) \sum_{\boldsymbol{y}\in\mathcal{Y}^k} p(z|\boldsymbol{y})p(\boldsymbol{y}|\boldsymbol{x}) \tag{5.43}$$

$$\cdot \log \frac{\sum_{z\in\mathcal{Z}} p(\tilde{z}|z) \sum_{\boldsymbol{y}\in\mathcal{Y}^k} p(z|\boldsymbol{y})p(\boldsymbol{y}|\boldsymbol{x})}{\sum_{\boldsymbol{x}'\in\mathcal{X}^l} p(\boldsymbol{x}') \sum_{z\in\mathcal{Z}} p(\tilde{z}|z) \sum_{\boldsymbol{y}\in\mathcal{Y}^k} p(z|\boldsymbol{y})p(\boldsymbol{y}|\boldsymbol{x}')} \qquad \text{subject to} \quad |\mathcal{Z}| = n.$$

We note that (5.43) is a convex maximization problem. Hence, the optimal quantizer is deterministic, i.e., we have $p^\star(z|\boldsymbol{y}) \in \{0, 1\}$ (cf. Proposition 5.2). Furthermore, the labels of the quantizer outputs enter in the objective function in (5.43) through $p(\tilde{z}|z)$. Therefore, the solution of (5.43) consists of an optimal partition of $\mathcal{Y}^k$ together with the corresponding optimal labels for the n quantizer outputs.

Depending on the forward channel, the channel-optimized vector quantizer may leave some of the n quantizer outputs unused, i.e., we may have $p(z) = 0$ for some $z \in \mathcal{Z}$. This is in contrast to conventional VQ where we have $p(z) > 0$ for all $z \in \mathcal{Z}$. We have formulated the COVQ problem in terms of DMCs. If the unquantized channel is memoryless with continuous output, the algorithm presented below can still be applied by discretizing y to the required precision. The extension to continuous-output forward channels is more difficult, since $p(\tilde{z}|z)$ may depend on $p(z)$, e.g., through an average power constraint, but $p(z)$ is not known *a priori*. For an error-free forward channel, (5.41) is equivalent to

$$p^\star(z|\boldsymbol{y}) = \arg\max_{p(z|\boldsymbol{y})} I(\mathsf{x};\mathsf{z}) \qquad \text{subject to} \quad |\mathcal{Z}| = n, \tag{5.44}$$

i.e., to vector quantizer design for maximum mutual information. We note that even for the simpler problem in (5.44), there exists in general no efficient algorithm for finding a globally optimal solution.

We next develop an algorithm that is based on the IB method (cf. Section 2.7) and yields a locally optimal solution of (5.43). To this end, we make the following major modifications compared to the basic IB algorithm (cf. Algorithm 2.2):

- We include the forward channel $p(\tilde{z}|z)$ into the quantizer optimization and we adapt the iterative algorithm accordingly.
- Since we know that the optimal quantizer is deterministic, we ensure that the output of the algorithm corresponds to a deterministic quantizer.
- We let the trade-off parameter $\beta \to \infty$ since we are interested in preserving as much relevant information as possible at a given quantization rate.

We first rewrite the objective function in (5.41) as follows:

$$\begin{aligned} I(\mathsf{x};\tilde{\mathsf{z}}) &= I(\mathsf{x};\mathsf{z}) + \underbrace{I(\mathsf{x};\tilde{\mathsf{z}}|\mathsf{z})}_{=0} - I(\mathsf{x};\mathsf{z}|\tilde{\mathsf{z}}) && (5.45)\\ &= I(\mathsf{x};\mathsf{y}) - I(\mathsf{x};\mathsf{y}|\mathsf{z}) - I(\mathsf{x};\mathsf{y}|\tilde{\mathsf{z}}) + I(\mathsf{x};\mathsf{y}|\mathsf{z},\tilde{\mathsf{z}}) && (5.46)\\ &= I(\mathsf{x};\mathsf{y}) - I(\mathsf{x};\mathsf{y}|\tilde{\mathsf{z}}). && (5.47) \end{aligned}$$

Since the first term in (5.47) does not depend on $p(z|\boldsymbol{y})$, we can further rewrite (5.41) as

$$p^\star(z|\boldsymbol{y}) = \operatorname*{arg\,min}_{p(z|\boldsymbol{y})} I(\mathsf{x};\mathsf{y}|\tilde{\mathsf{z}}) = \operatorname*{arg\,min}_{p(z|\boldsymbol{y})} \mathbb{E}\big\{\mathbb{E}\{C(\mathsf{y},\tilde{\mathsf{z}})|\mathsf{y}\}\big\} \qquad \text{subject to} \quad |\mathcal{Z}| = n, \tag{5.48}$$

where we have defined

$$C(\boldsymbol{y},\tilde{z}) \triangleq D\big(p(\boldsymbol{x}|\boldsymbol{y})\big\|p(\boldsymbol{x}|\tilde{z})\big)\,. \tag{5.49}$$

The conditional expectation in (5.48) can be written as follows:

$$\mathbb{E}\{C(\mathsf{y},\tilde{\mathsf{z}})|\mathsf{y}{=}\boldsymbol{y}\} = \sum_{\tilde{z}\in\tilde{\mathcal{Z}}} p(\tilde{z}|\boldsymbol{y})C(\boldsymbol{y},\tilde{z}) \tag{5.50}$$

$$= \sum_{z\in\mathcal{Z}} p(z|\boldsymbol{y}) \sum_{\tilde{z}\in\tilde{\mathcal{Z}}} p(\tilde{z}|z)C(\boldsymbol{y},\tilde{z}). \tag{5.51}$$

We next choose $p(z|\boldsymbol{y})$ such that $\mathbb{E}\{C(\mathsf{y},\tilde{\mathsf{z}})|\mathsf{y}{=}\boldsymbol{y}\}$ is minimized for each $\boldsymbol{y}\in\mathcal{Y}^k$. To this end, we note that the second sum in (5.51) is a constant for fixed z. Hence, we let

$$p(z|\boldsymbol{y}) = \delta_{z,z^\star(\boldsymbol{y})}, \tag{5.52}$$

where $z^\star(\boldsymbol{y})$ is the particular $z\in\mathcal{Z}$ which minimizes the second sum in (5.51). We thus have

$$z^\star(\boldsymbol{y}) = \operatorname*{arg\,min}_{z\in\mathcal{Z}} \sum_{\tilde{z}\in\tilde{\mathcal{Z}}} p(\tilde{z}|z)C(\boldsymbol{y},\tilde{z}). \tag{5.53}$$

By minimizing (5.51) for each $\boldsymbol{y}\in\mathcal{Y}^k$ separately, we also minimize the objective function of (5.48) for fixed $C(\boldsymbol{y},\tilde{z})$. Furthermore, due to (5.52) we have

$$p(\tilde{z}|\boldsymbol{y}) = \sum_{z\in\mathcal{Z}} p(\tilde{z}|z)p(z|\boldsymbol{y}) = p\big(\tilde{z}|z^\star(\boldsymbol{y})\big). \tag{5.54}$$

The above expressions allow us to formulate our algorithm for channel-optimized vector quantizer design, see Algorithm 5.4. The proposed algorithm terminates if the relative decrease in the objective function value between two iterations is below a prescribed threshold or if a certain number of iterations has been performed. Algorithm 5.4 can be used for the design of conventional vector quantizers by setting $p(\tilde{z}|z) = \delta_{\tilde{z},z}$. Furthermore, the design of scalar quantizers is included as the special case where $k = 1$. The convergence of the proposed algorithm to a locally optimal solution of (5.43) is guaranteed by the IB method. We note that algorithm 5.4 can be run repeatedly with the best solution retained; this helps to avoid getting stuck in a bad local optimum.

We emphasize that our algorithm finds the optimal quantizer *jointly* with corresponding labels for the quantizer output. This is in contrast to distortion-based channel-optimized vector quantizer design algorithms, which usually require that the labels are fixed in advance. Hence, in our case the NP-hard label optimization problem is avoided and need not be considered separately.

Algorithm 5.4 *Channel-optimized vector design algorithm.*

Input: $\mathcal{X}^l$, $\mathcal{Y}^k$, $\mathcal{Z}$, $\tilde{\mathcal{Z}}$, $p(\boldsymbol{x},\boldsymbol{y})$, $p(\tilde{z}|z)$, $\varepsilon > 0$, $M \in \mathbb{N}$

Initialization: $\overline{C}^{(0)} \leftarrow \infty$, $\eta \leftarrow \infty$, $m \leftarrow 1$, randomly initialize $C(\boldsymbol{y},\tilde{z}) \in \mathbb{R}_+, \forall \boldsymbol{y} \in \mathcal{Y}^k$ and $\forall \tilde{z} \in \tilde{\mathcal{Z}}$

1: **while** $\eta \geq \varepsilon$ **and** $m \leq M$ **do**
2: **for all** $\boldsymbol{y} \in \mathcal{Y}^k$ **do**
3: $z^\star \leftarrow \arg\min_{z\in\mathcal{Z}} \sum_{\tilde{z}\in\tilde{\mathcal{Z}}} p(\tilde{z}|z) C(\boldsymbol{y},\tilde{z})$
4: $p^{(m)}(z|\boldsymbol{y}) \leftarrow \delta_{z,z^\star},\ z \in \mathcal{Z}$
5: $p(\tilde{z}|\boldsymbol{y}) \leftarrow p(\tilde{z}|z^\star),\ \tilde{z} \in \tilde{\mathcal{Z}}$
6: **end for**
7: $p(\tilde{z}) \leftarrow \sum_{\boldsymbol{y}\in\mathcal{Y}^k} p(\tilde{z}|\boldsymbol{y}) p(\boldsymbol{y})$
8: $p(\boldsymbol{x}|\tilde{z}) \leftarrow \frac{1}{p(\tilde{z})} \sum_{\boldsymbol{y}\in\mathcal{Y}^k} p(\boldsymbol{x},\boldsymbol{y}) p(\tilde{z}|\boldsymbol{y})$
9: $C(\boldsymbol{y},\tilde{z}) \leftarrow D\Big(p(\boldsymbol{x}|\boldsymbol{y}) \Big\| p(\boldsymbol{x}|\tilde{z})\Big)$
10: $\overline{C}^{(m)} \leftarrow \sum_{\boldsymbol{y}\in\mathcal{Y}^k} p(\boldsymbol{y}) \sum_{\tilde{z}\in\tilde{\mathcal{Z}}} p(\tilde{z}|\boldsymbol{y}) C(\boldsymbol{y},\tilde{z})$
11: $\eta \leftarrow (\overline{C}^{(m-1)} - \overline{C}^{(m)})/\overline{C}^{(m)}$
12: $m \leftarrow m + 1$
13: **end while**

Output: channel-optimized quantizer $p^{(m-1)}(z|\boldsymbol{y})$

5.6 Comparison of Algorithms and Application Examples

In this section, we compare the proposed algorithms and provide application examples. In particular, we give guidelines for the selection of a quantizer design algorithm, we study the convergence behavior of the proposed algorithms, we compare scalar quantizers to the information-theoretic limit, and we compare mutual-information-optimal quantization to MSE-optimal quantization. Furthermore, we give two numerical examples which respectively consider low-density parity-check (LDPC) decoding with quantized LLRs and channel-optimized quantization for receivers with unreliable memory.

5.6.1 Algorithm Comparison

We first give some guidelines for the choice of the appropriate quantizer design algorithm. If channel-optimized quantization or VQ is required, then Algorithm 5.4 may be used. Furthermore, Algorithm 5.4 is suitable in case the optimal quantization regions are nonconvex. If y is a continuous random variable, the application of Algorithm 5.4 requires discretization of y to the desired precision. Algorithms 5.1 and 5.2 are well suited for scalar quantizer design when the optimal quantization regions are intervals. In the special case of LLR quantization, Algorithm 5.2 does not require root-finding and is guaranteed to converge to a local optimum. The greedy approach of Algorithm 5.3 is a simple and useful alternative to Algorithms 5.1 and 5.2 when root-finding methods should be avoided.

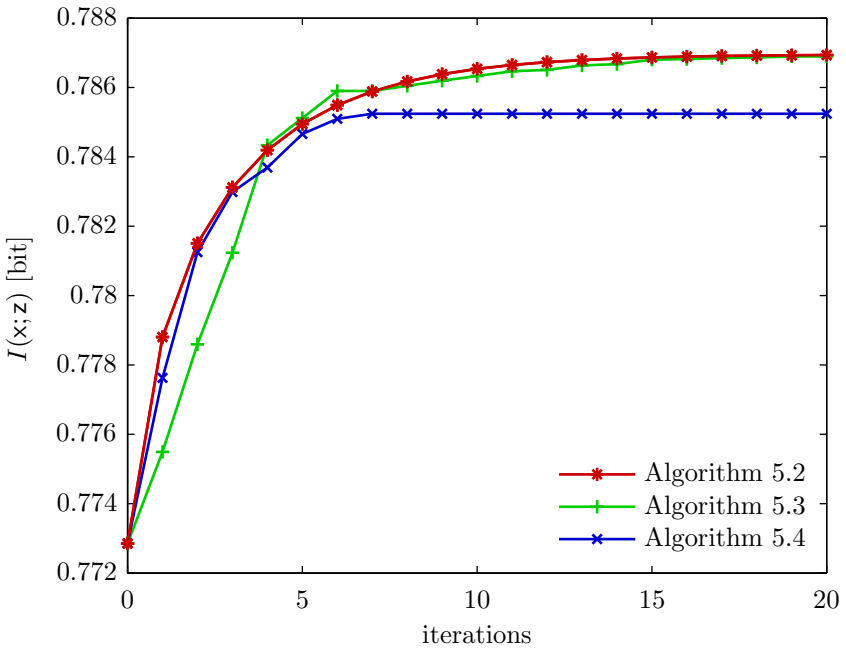


Figure 5.3: Convergence behavior of the proposed algorithms for scalar quantization of conditionally Gaussian LLRs.

We next compare the convergence behavior of the proposed algorithms. To this end, we consider the quantization of conditionally Gaussian LLRs, i.e., we have $\mathsf{y}|x \sim \mathcal{N}(x\mu, 2\mu)$ with $x \in \{-1, 1\}$ and $\mu > 0$. Figure 5.3 shows the value of the objective function $I(\mathsf{x};\mathsf{z})$ versus the iteration number for the Algorithms 5.2-5.4. Here, we use $n = 8$ quantization levels and we have chosen $\mu = 5$. We have initialized all algorithms using the MOE quantizer. Algorithms 5.2 (red curve, '$*$' markers) and 5.3 (green curve, '$+$' markers) converge to the optimal quantizer within 20 iterations. The greedy algorithm has performed one refinement step after 6 iterations. We have observed that in this setting $I(\boldsymbol{g})$ is strictly quasiconcave in $\boldsymbol{g}$. However, we were unable to formally verify this observation. For a strictly quasiconcave objective function $I(\boldsymbol{g})$, the Algorithms 5.2 and 5.3 converge to the globally optimal quantizer. Furthermore, we observe that Algorithm 5.4 (blue curve, '$\times$' markers) gets stuck in a local optimum and does not find the globally optimal quantizer.

In Figure 5.4, we show how the convergence of Algorithm 5.4 is influenced by the initialization. The dashed line and the dotted line respectively show the best and the worst result for 10^4 random initializations. The solid line corresponds to the MOE initialization as in Figure 5.3. We observe that the best initialization yields the optimal quantizer within 3 iterations. The gap between the best case and the worst case in terms of $I(\mathsf{x};\mathsf{z})$ is approximately 1.26 %. Similarly, the MOE initialization is 0.2 % away from the optimal quantizer in terms of mutual information. Hence, although the MOE initialization does not yield the globally optimal quantizer in this case, it is a suitable choice for the initialization of Algorithm 5.4.

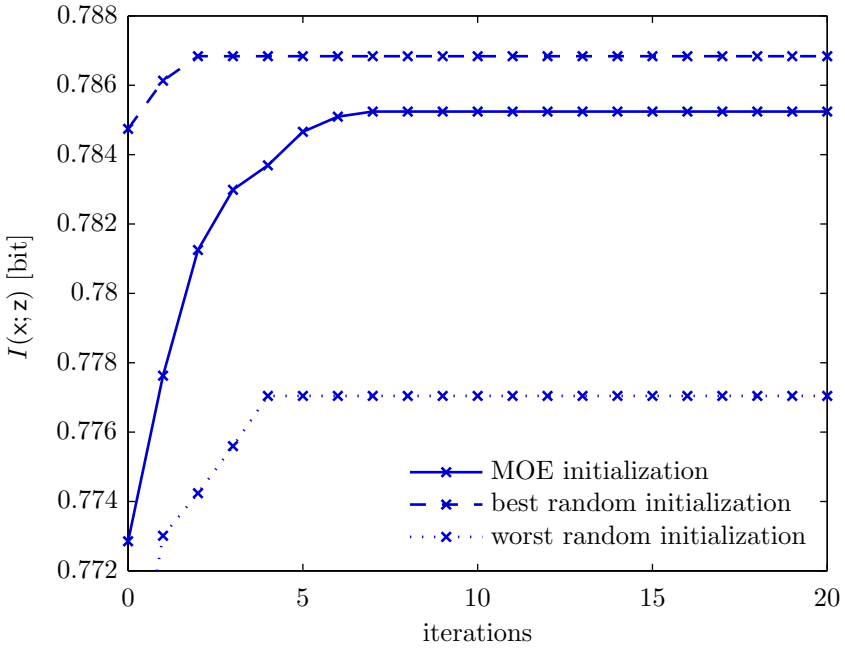


Figure 5.4: Dependence of the convergence behavior of Algorithm 5.4 on the initialization.

Next, we compare the performance of scalar quantizers to the information-theoretic limit in the same setting as above (conditionally Gaussian LLRs with $\mu \in \{1, 5, 10\}$). The solid lines in Figure 5.5 show the optimal rate-information trade-off (cf. Section 4.2 for a formal definition of the rate-information trade-off) for the different values of μ which we have computed using the IB algorithm (cf. Algorithm 2.2). These curves tend to the respective value of $I(\mathsf{x};\mathsf{y})$ as the quantization rate R becomes large. We have designed scalar quantizers with $2, \ldots, 8$ quantization levels using Algorithm 5.2. The rate-information pairs achieved by these quantizers are indicated by the '$\times$' markers. Note that $R = H(\mathsf{z})$ since the quantizers are deterministic. We observe that the quantizers closely approach the optimal rate-information trade-off. In particular, for $\mu = 5$ each quantizer is less than 1 % away from the optimal rate-information trade-off. Therefore, VQ can only provide a negligible gain in terms of performance over scalar quantization. The main advantage of VQ in this setting is the increased flexibility regarding the quantization rate. Furthermore, time-sharing can be used to (asymptotically) achieve all points on a line connecting the rate-information pairs corresponding to two quantizers.

Figure 5.5 moreover shows the performance of MSE-optimal quantizers with $2, \ldots, 8$ quantization levels ('+' markers). In contrast to the jointly Gaussian case (cf. Section 4.7), MSE-optimal quantization is inferior in the setting we consider here. For small μ (corresponding to low signal-to-noise ratio (SNR)) the difference between MSE-optimal quantization and mutual-information-optimal quantization is rather small. However, as μ increases the suboptimality of MSE-optimal quantizers in terms of the rate-information trade-off becomes more pronounced.

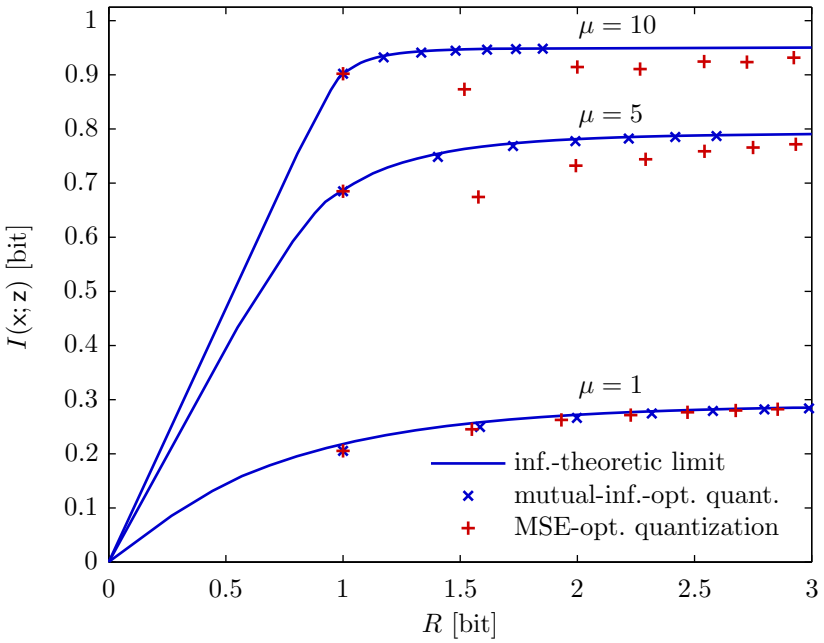


Figure 5.5: Comparison of scalar quantizers with 2 to 8 quantization levels to the information-theoretic limit.

In Figure 5.6, we show how the rate and the mutual information $I(\mathsf{x};\mathsf{z})$ depend on the position of the quantizer boundaries (solid lines) for $\mu \in \{5, 10\}$. In particular, we consider quantizers with 3 and 4 quantization levels. For reasons of symmetry, the optimal quantizer is always symmetric in the considered setting. Hence, there is only one free parameter when we consider 3 and 4 quantization levels. The '$\times$' markers correspond to the respective mutual-information-optimal quantizers and the dashed lines correspond to the optimal rate-information trade-off. We observe that Algorithm 5.2 indeed finds the global optimum in these cases since all markers are at the maximum of the respective solid line. The rate-information pairs achieved by the quantizers are very close to the optimal rate-information trade-off. However, in all cases a different quantizer boundary position yields a quantizer which comes even closer to the optimal trade-off. Unfortunately, it is unclear how to formulate and solve an optimization problem for quantizer design such that it yields a quantizer which is as close as possible to the optimal trade-off for a fixed number of quantization levels. Figure 5.6 also shows that the MOE quantizers are substantially worse in this setting than the mutual-information-optimal quantizers.

In Figure 5.7, we plot $p(y)$ for $y > 0$ and $\mu = 5$ together with the mutual-information-optimal quantizer and the MSE-optimal quantizer for 4 quantization levels. In this case the quantizers are symmetric, i.e., $\boldsymbol{g}$ is of the form $(-g \;\; 0 \;\; g)^{\mathrm{T}}$. The dashed lines show the positive quantizer boundary for the mutual-information-optimal quantizer and the MSE-optimal quantizer, respectively. The '$\times$' and '$+$' markers show the quantized LLRs corresponding to

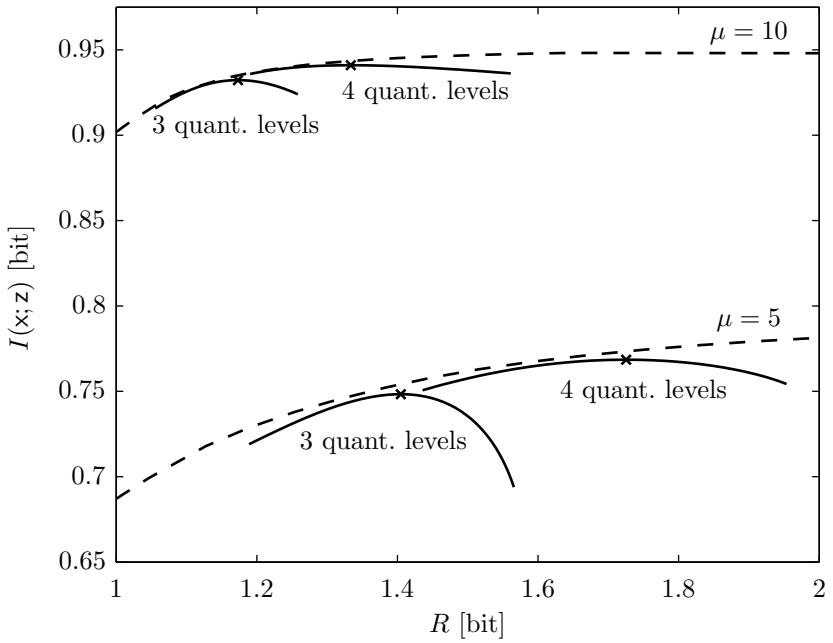


Figure 5.6: Behavior of the rate and the mutual information $I(\mathsf{x};\mathsf{z})$ as the quantizer boundaries vary. Mutual-information-optimal quantizers are indicated by '$\times$' markers and the dashed lines correspond to the optimal rate-information trade-off.

the two quantizers. This shows that there is a substantial difference between the mutual-information-optimal quantizer and the MSE-optimal quantizer.

5.6.2 Application Examples

LDPC Decoding with Quantized LLRs. We consider channel-coded data transmission over a binary-input additive white Gaussian noise (AWGN) channel using the rate-1/2 DVB-S2 LDPC code with a blocklength of 64800 bits. The receiver uses the channel output to compute LLRs for the code bits which are then quantized. A belief propagation (BP) decoder with 40 iterations finally decodes the transmitted data based on the quantized LLRs.

Figure 5.8 shows the bit error rate (BER) versus SNR performance for LLR quantization with $2, \ldots, 8$ quantization levels and 40 decoder iterations. For comparison, we also plot the BER performance in the unquantized case ('+' marker). The quantizers have been designed using Algorithm 5.2. We observe that each additional quantization level yields a smaller performance improvement with increasing resolution of the quantizer. The SNR penalty compared to the unquantized case is 1.6 dB for 2 quantization levels, 0.4 dB for 4 quantization levels, and 0.1 dB for 8 quantization levels.

In Figure 5.9, we study the influence of the optimality criterion in the quantizer design on the BER performance. Specifically, we compare mutual-information-optimal quantization (solid lines) to MSE-optimal quantization (dashed lines) for 2, 4, and 8 quantization levels.

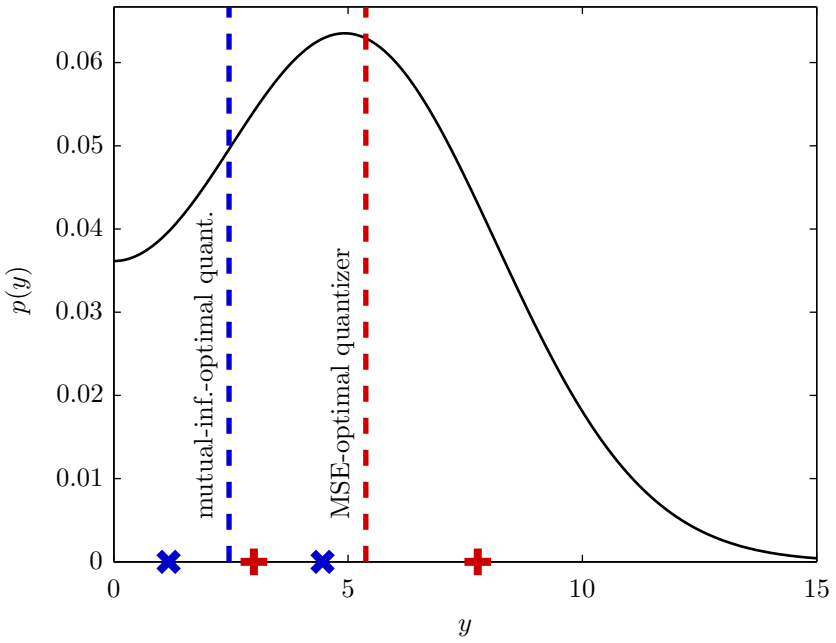


Figure 5.7: Comparison between mutual-information-optimal quantizer and MSE-optimal quantizer for 4 quantization levels.

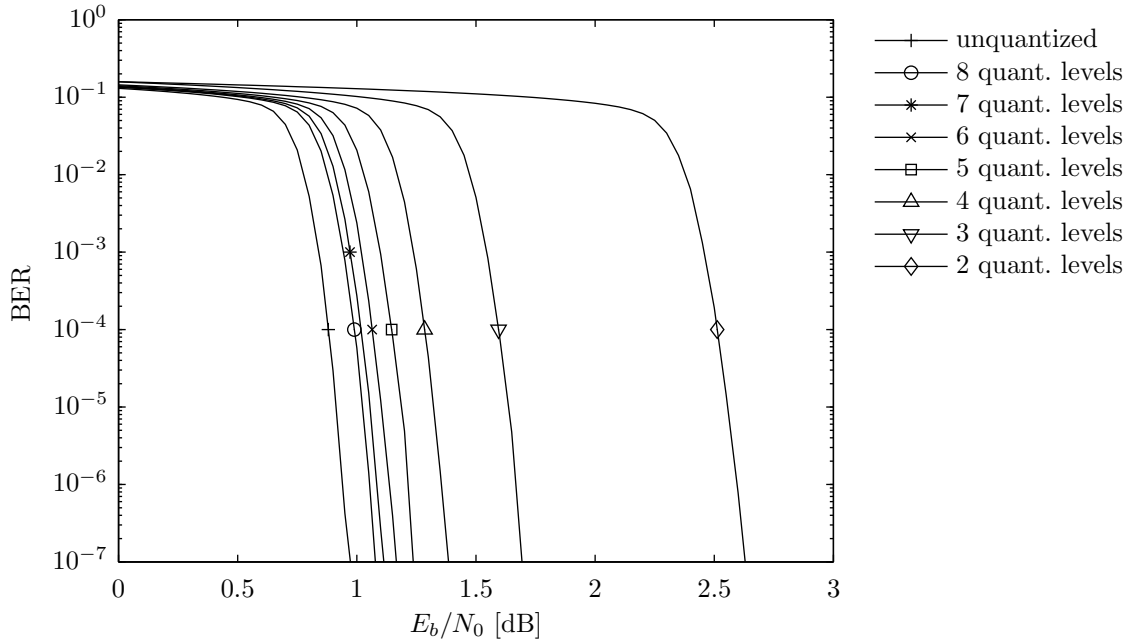


Figure 5.8: BER performance of the rate-1/2 DVB-S2 LDPC code (blocklength 64800 bits) with quantized LLRs and 40 BP decoder iterations.

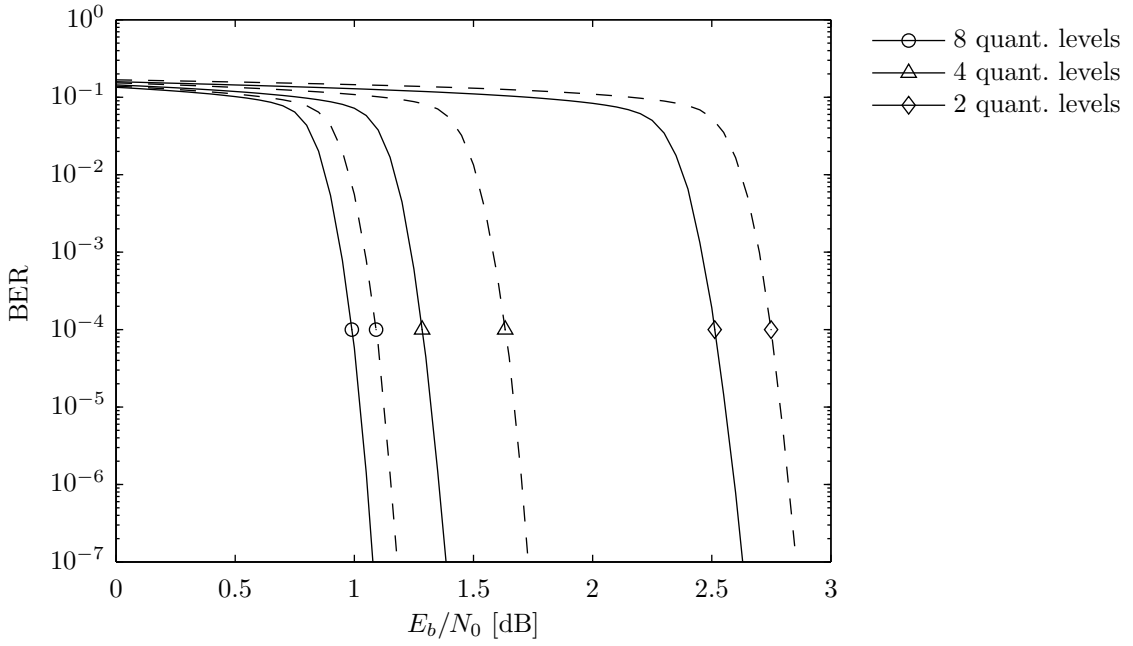


Figure 5.9: Influence of the quantizer design on the BER performance of the rate-1/2 DVB-S2 LDPC code (blocklength 64800 bits). Solid lines correspond to mutual-information-optimal quantizers and dashed lines correspond to MSE-optimal quantizers.

It turns out that MSE-optimal quantization is inferior to mutual-information-optimal quantization by 0.23 dB for 2 quantization levels, 0.34 dB for 4 quantization levels, and 0.1 dB for 8 quantization levels. We note that this additional SNR penalty due to an inappropriate quantizer design is significant since the considered LDPC code operates close to capacity.

Receivers with Unreliable Memory and Channel-Optimized LLR quantization. We next consider the setting depicted in Figure 5.10. Here, binary data $\mathsf{x} \in \{-1, 1\}$ is transmitted over the AWGN channel $\mathsf{y}' = \mathsf{x} + \mathsf{w}$, where $\mathsf{w} \sim \mathcal{N}(0, \sigma^2)$ is independent of x. The receiver uses the channel output y' to calculate the LLR $y = 2y'/\sigma^2$ which is then quantized with n quantization levels. Next, the quantizer output $z = q(y)$ is mapped to a binary label $\boldsymbol{b} = \phi(z) \in \{0, 1\}^{\lceil \log_2 n \rceil}$ which is stored in unreliable memory. Reading the data from the memory yields the possibly corrupted label $\tilde{\boldsymbol{b}}$ which corresponds to the quantizer output $\tilde{z} = \phi^{-1}(\tilde{\boldsymbol{b}})$. We use the stuck-at channel (SAC) to model the failure of bit cells in the unreliable memory [89]. For the SAC with error probability $0 < \varepsilon < 1$ and equally likely stuck-at errors, each bit cell is in one of the following three states:

- $\Xi = \xi_0$: error-free bit cell (with probability $1 - \varepsilon$),
- $\Xi = \xi_+$: bit cell is stuck at "1" (with probability $\varepsilon/2$),
- $\Xi = \xi_-$: bit cell is stuck at "0" (with probability $\varepsilon/2$).

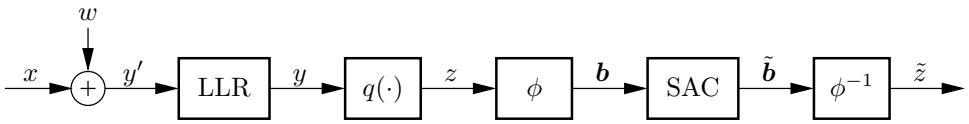


Figure 5.10: Receiver with unreliable LLR memory. The channel-optimized quantizer $q(\cdot)$ maximizes $I(\mathsf{x};\tilde{\mathsf{z}})$.

The capacity of a single bit cell in the SAC model is zero since the content of the bit cell is independent of the input with positive probability. However, we use the SAC to model unreliable memory in the following way: we assume that each bit cell fails independently and data is stored without knowledge about the state of the individual bit cells. This is equivalent to multiple channel uses of a single bit cell where the state of the bit cell is chosen at random before each channel use. In this case, the SAC with error probability ε and equally likely stuck-at errors is equivalent to a binary symmetric channel with crossover probability $\varepsilon/2$. Indeed, we have

$$p(\tilde{b}|b) = \sum_{\xi\in\{\xi_0,\xi_+,\xi_-\}} p(\tilde{b}|b,\xi)p(\xi) \tag{5.55}$$

$$= (1-\varepsilon)\delta_{\tilde{b},b} + \frac{\varepsilon}{2}\delta_{\tilde{b},0} + \frac{\varepsilon}{2}\delta_{\tilde{b},1} \tag{5.56}$$

$$= \begin{cases} 1-\frac{\varepsilon}{2}, & b=\tilde{b} \\ \frac{\varepsilon}{2}, & b\neq\tilde{b} \end{cases}, \tag{5.57}$$

where $b,\tilde{b}\in\{0,1\}$.

In what follows, we assume $\varepsilon = 0.11003$ and we perform channel-optimized scalar quantization with $n = 8$ quantization levels. The channel-optimized quantizers are designed using Algorithm 5.4. In Figure 5.11, we plot the mutual information $I(\mathsf{x};\tilde{\mathsf{z}})$ versus the SNR $1/\sigma^2$ in dB. The red curve ('$\circ$' markers) shows the rates achievable by our channel-optimized quantizer design. The blue curve ('+' markers) constitutes a simple upper bound given by error-free storage of the quantizer output. The solid (dashed) green curve ('$*$' markers) shows the rates achievable by non-channel-optimized quantization maximizing $I(\mathsf{x};\mathsf{z})$ using the best (worst) mapping $\phi(\cdot)$. We note that in this case there are $8! = 40320$ different bit mappings and the performance penalty may be very large if the optimal mapping is not found. This is in contrast to our approach which outperforms non-channel-optimized quantization without the need to perform separate optimization of the bit labels.

5.7 Discussion

In this chapter, we have studied mutual-information-optimal quantizer design for communication problems. We have proposed an alternating optimization algorithm and a greedy algorithm for the design of scalar quantizers that maximize mutual information. These al-

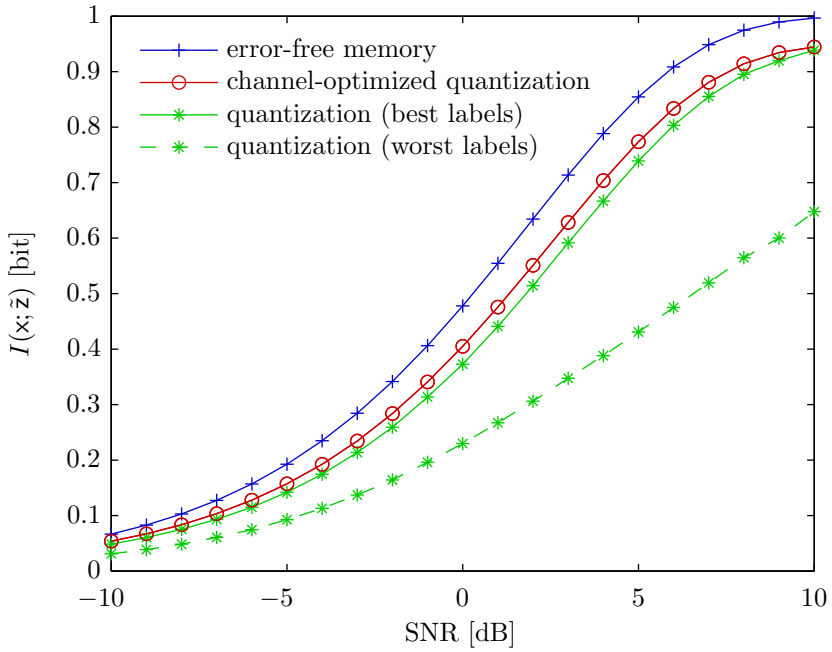


Figure 5.11: Comparison of $I(\mathsf{x};\tilde{\mathsf{z}})$ for receiver processing with unreliable LLR memory.

gorithms are simple to implement and exhibit excellent convergence behavior. Furthermore, we have proposed an algorithm for the design of channel-optimized vector quantizers. This algorithm finds a locally optimal quantizer together with the labels for the quantizer output and therefore avoids the NP-hard label optimization problem. We have found that the MOE quantizer is a suitable initialization for the proposed quantizer design algorithms. Our numerical results show that the performance of scalar LLR quantizers closely approaches the information-theoretic limit. This implies that VQ can only provide a negligibly small performance improvement over scalar quantization. Moreover, in contrast to the jointly Gaussian case (cf. Section 4.7), MSE-optimal quantization is inferior to mutual-information-optimal quantization in terms of the rate-information trade-off.

In the application examples of Section 5.6 the receiver first computes LLRs with full resolution and then quantizes the LLRs. This can be avoided by mapping the quantization regions for the LLRs to the corresponding quantization regions for the channel output. We are thus able to obtain quantized LLRs directly from the channel output which is of course more efficient than computing unquantized LLRs first. An extension of the alternating optimization algorithm of Section 5.3 to quantizer design based on training data may be possible. This would enable online quantizer design without the need for knowledge of the joint distribution $p(x, y)$.

6

Quantization-Based Network Coding for the MARC

In this chapter, we consider relay-based cooperative communication which allows us to apply results from Chapter 5 and Chapter 3. Specifically, we present a transmission scheme for the multiple-access relay channel (MARC) which incorporates network coding [2] at the physical layer and allows for a low-complexity implementation of the processing at the relay. Section 6.1 introduces the basic idea of the proposed transmission scheme and gives background information on related work. In Section 6.2, the system model for the MARC with n sources is presented and the considered channel models are described. Next, the basic operation of all network nodes is discussed in Section 6.3. In Section 6.4, we explain the relay processing, which essentially consists of log-likelihood ratio (LLR) quantization followed by network encoding, in detail. The destination decodes the source data using an iterative turbo-like joint network-channel decoder which is presented in Section 6.5. We numerically evaluate the performance of the proposed transmission scheme in Section 6.6 using Monte Carlo simulations. The discussion in Section 6.7 concludes this chapter.

6.1 Introduction and Background

The MARC extends the classical relay channel [19] and models data transmission of multiple sources to a common destination with the help of one relay (cf. Figure 6.1). In this setting, the purpose of the relay is to facilitate the transmission of the sources by providing cooperative diversity [92,93]. Application examples include, but are not limited to, the cooperative uplink in cellular systems and wireless sensor networks with data transmission to a fusion center.

Network coding [2] allows intermediate network nodes to combine data flows and is well known for its ability to increase throughput and reliability. These benefits have motivated the study of transmission schemes for the MARC which incorporate network coding at the physical layer [16, 41, 42, 46, 74, 101, 108, 109, 113, 115, 118, 119]. Achievable rates and outer bounds for the capacity region of the MARC have been established in [52]. We note that the capacity region of the MARC is unknown; in fact, even the capacity of the (nondegraded) relay channel is unknown.

Decode-and-forward (DF) schemes with network coding for the MARC with two sources and orthogonal channels have been proposed in [16, 42]. In these schemes, the relay decodes the source messages individually and forwards a network-coded combination of the data to the destination. Iterative decoding (cf. Subsection 2.6.5) is used at the destination to jointly decode the channel codes and the network code. A disadvantage of DF-based schemes is that the relay is required to fully decode the source messages which in turn requires the relay to be close to the sources. Furthermore, performing channel decoding at the relay adds complexity and delay to the system. Extensions of [42] to more than two sources and to the MARC with simultaneous multiple-access are studied in [46] and [41], respectively.

In contrast to DF-based schemes which perform finite-field network encoding of the decoded messages, the analog network coding schemes in [74, 101, 115] compute a many-to-one function of the analog signals received at the relay. In the MARC with simultaneous multiple-access, network encoding is essentially performed by the channel due to the interfering source transmissions. The idea of exploiting the wireless channel for physical layer network coding has been proposed independently and concurrently in [75, 83, 121]. While [83, 121] focus on two-way relaying scenarios, [75] has evolved into the more general compute-and-forward scheme [76, 77] which uses nested lattice codes (cf. the survey paper [116]) to decode a set of linear combinations of the source messages.

The basic idea we follow in this chapter has been introduced in [113] where the relay performs "soft combining" and forwards LLRs for the network-coded bits. This approach is able to exploit the information which is available at the relay without requiring that the relay (fully) decodes the source messages. Hence, this scheme is well suited for unreliable source-relay channels and practical channel codes with finite blocklengths. The work in [118] augments [113] with optimized scalar quantization at the relay and thus avoids analog LLR forwarding. Two-dimensional vector quantization is used in [119] instead of soft combining with subsequent scalar quantization, yielding improved performance in asymmetric channel

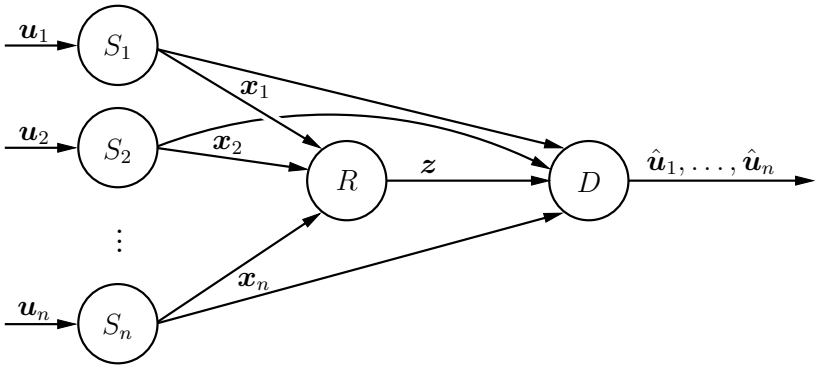


Figure 6.1: The MARC with n sources.

conditions. In [118, 119] the information bottleneck method [97] has first been applied in the communications context for quantizer design. Our previous work [108] studies efficient encoding for the scheme in [119] which enables the extension to more than two sources [109]. In this chapter, we extend [109] by modeling the relay-destination link as a noisy channel and by performing channel-optimized quantization at the relay. Furthermore, in contrast to [108, 109, 118, 119] the transmission scheme proposed in this chapter does not perform channel decoding at the relay.

6.2 System Model

In this section we introduce the basic model for the MARC and the links between the individual nodes.

6.2.1 MARC Model

We consider the time-division MARC with n sources, $S_1, S_2, \ldots, S_n$, one half-duplex relay R, and a destination D as depicted in Figure 6.1. The assumptions of orthogonal channels and half-duplex nodes simplify practical implementation. The sources consecutively broadcast their independent messages in the first n time slots. In the $(n+1)$th time slot, the relay forwards to the destination a suitably compressed version of the data it has received in the previous n time slots (cf. Section 6.4). Finally, the destination jointly decodes all signals received in the $n+1$ time slots (cf. Section 6.5). We note that in our model the sources do not overhear each others transmission. In what follows, we assume that the total transmission time is shared equally among all sources and the relay, i.e., each transmitting node uses the channel M times per time slot and, hence, there are $(n+1)M$ channel uses in total (optimization of the resource allocation is beyond the scope of this work). It is important to note that the relay's transmission causes a rate loss which becomes smaller as n increases. Next, we describe the channel models for the individual links.

6.2.2 Channel Models

Source-Relay and Source-Destination Channels. We use a quasi-static channel model with pathloss and additive white Gaussian noise for the source-relay and source-destination links. The transmissions to the relay and the destination take place on orthogonal channels. Therefore, we have the following input-output relation (the vectors in (6.1) are of length M):

$$\mathbf{y}_{i,j} = d_{i,j}^{-\alpha/2}\mathsf{h}_{i,j}\mathbf{x}_i + \mathbf{w}_{i,j}, \quad i \in \{1,2,\ldots,n\},\ j \in \{R,D\}, \tag{6.1}$$

where $\mathbf{x}_i$ is the signal transmitted by the ith source, $\mathbf{y}_{i,j}$ is the corresponding receive signal at node j (either the relay or the destination), $d_{i,j}$ is the distance between nodes i and j, α is the path-loss exponent, $\mathbf{w}_{i,j} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\boldsymbol{I})$ is circularly symmetric white Gaussian noise with variance σ^2, and $\mathsf{h}_{i,j}$ denotes the gain of the channel from node i to node j. Throughout we assume that time-division is used to achieve orthogonal transmissions. This is, however, not a requirement of the proposed scheme; in fact, any multiple-access scheme yielding an input-output relation as in (6.1) can be employed, e.g., orthogonal frequency-division multiple access or code division multiple access.

For the sake of simplicity, we assume single-carrier transmissions over nondispersive channels. However, the proposed network coding scheme operates on the bit-level and can thus also be used with multi-carrier modulation formats over frequency-selective channels (the relay operations are then performed on a per-subcarrier basis). In what follows, we impose a transmit energy constraint for the sources, i.e., we fix $E_s \triangleq \mathbb{E}\{\|\mathbf{x}_i\|_2^2\}$, $i = 1,2,\ldots,n$. Such a constraint is reasonable, especially for mobile devices and battery-powered sensors. The signal-to-noise ratio (SNR) of the link between node i and node j for the channel realization $h_{i,j}$ is given by

$$\gamma_{i,j} = |h_{i,j}|^2 \frac{d_{i,j}^{-\alpha} E_s}{M\sigma^2}, \tag{6.2}$$

and the average SNR equals

$$\overline{\gamma}_{i,j} = \mathbb{E}\{|\mathsf{h}_{i,j}|^2\} \frac{d_{i,j}^{-\alpha} E_s}{M\sigma^2}. \tag{6.3}$$

We assume that each node has receive channel state information (CSI) only, i.e., the $h_{i,j}$'s are known at node j. This implies in particular that the relay has no CSI about the source-destination channels. In Section 6.6, we use the channel model (6.1) with $\mathsf{h}_{i,j} \sim \mathcal{CN}(0,1)$ (corresponding to a frequency-flat Rayleigh fading channel with $\overline{\gamma}_{i,j} = d_{i,j}^{-\alpha} E_s/(M\sigma^2)$) and with $\mathsf{h}_{i,j} = 1$ (corresponding to a constant additive white Gaussian noise (AWGN) channel with $\overline{\gamma}_{i,j} = \gamma_{i,j}$).

Relay-Destination Channel. We model the relay-destination link using a discrete memoryless channel (DMC) with input alphabet $\mathcal{Z}_R$ and output alphabet $\mathcal{Z}_D$. This DMC is specified by the $|\mathcal{Z}_D| \cdot |\mathcal{Z}_R|$ transition probabilities $p(z_D|z_R)$, $z_D \in \mathcal{Z}_D, z_R \in \mathcal{Z}_R$. We assume that $p(z_D|z_R)$ is known at the relay. The benefit of modeling the relay-destination link as a

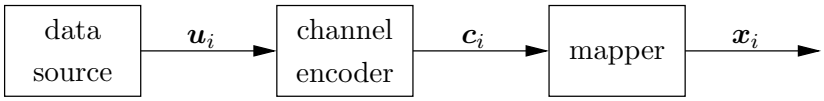


Figure 6.2: Block diagram of the ith source.

DMC is twofold. Firstly, it implicitly captures a rate-constraint since $\log_2\big(\min\{|\mathcal{Z}_R|, |\mathcal{Z}_D|\}\big)$ constitutes an upper bound on the number of bits that can be transmitted reliably per channel use. Secondly, the transition probabilities $p(z_D|z_R)$ can be used to model residual errors at the destination after channel decoding, i.e., the destination may not be able to recover z_R without error. We note that using a DMC to model the relay-destination link is not too restrictive, especially in the case of an operator-deployed relay.

6.3 Basic Node Operation

In this section, we describe the basic operation of all network nodes. The network encoding at the relay and the iterative decoding at the destination are discussed in more detail in Sections 6.4 and 6.5, respectively.

6.3.1 Sources

The ith source ($i \in \{1, 2, \ldots, n\}$) generates a length-$K_i$ sequence $\boldsymbol{u}_i \in \{0,1\}^{K_i}$ of independent and equally likely bits which has to be communicated to the destination. The data $\boldsymbol{u}_i$ is channel encoded using a linear binary code $\mathcal{C}_i$ of rate $R_i = K_i/N_i$, yielding the codeword $\boldsymbol{c}_i \in \{0,1\}^{N_i}$. Next, the code bits are mapped to a signal constellation $\mathcal{A}_i$ of cardinality $|\mathcal{A}_i| = 2^{m_i}$ which yields the transmit signal $\boldsymbol{x}_i \in \mathcal{A}^M$. The code rates are chosen as $R_i = K_i/(m_i M)$ to ensure that the transmission of each source requires M channel uses. For simplicity of exposition we let $K \triangleq K_i$ and $N \triangleq N_i$, $i = 1, \ldots, n$. Hence, we have $R_i = R$ and $m_i = m$, $i = 1, \ldots, n$. The sum rate (in bits per channel use) is then given by

$$R_s = \frac{K}{M}\frac{n}{n+1} = mR\frac{n}{n+1}. \tag{6.4}$$

If the relay was not present, the sum rate would be equal to $\tilde{R}_s = mR$ and thus the rate loss due to the half-duplex relay node equals

$$\Delta R_s = \tilde{R}_s - R_s = \frac{mR}{n+1}. \tag{6.5}$$

We note that the sources may use different channel codes and signal constellations. Furthermore, the proposed transmission scheme is not restricted to channel codes of equal dimension and blocklength. Figure 6.2 depicts the block diagram of a source.

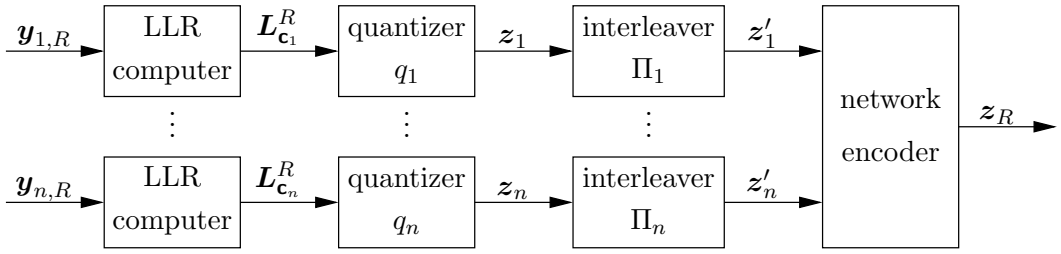


Figure 6.3: Block diagram of the relay.

6.3.2 Relay

In the proposed scheme, the relay performs LLR quantization with subsequent network encoding. This strategy has low computational complexity, which is important for practical implementations. In particular, the LLRs need not be computed with full resolution before quantization and the network encoding operation can be implemented using a simple table lookup operation. Figure 6.3 shows a block diagram of the relay which first computes the following LLRs for the code bits c_i:

$$L^R_{\mathsf{c}_{i,l}} = \log \frac{\mathbb{P}\left\{\mathsf{c}_{i,l}=0 \middle| \mathsf{y}^R_{i,\lceil l/m \rceil} = y^R_{i,\lceil l/m \rceil}\right\}}{\mathbb{P}\left\{\mathsf{c}_{i,l}=1 \middle| \mathsf{y}^R_{i,\lceil l/m \rceil} = y^R_{i,\lceil l/m \rceil}\right\}}, \qquad l = 1, \ldots, N, \quad i = 1, \ldots, n. \tag{6.6}$$

In (6.6), $\mathsf{c}_{i,l}$ denotes the lth code bit transmitted by the ith source, $y^R_{i,\lceil l/m \rceil}$ is the receive value corresponding to the symbol which carries $\mathsf{c}_{i,l}$, and the superscript "R" in $L^R_{\mathsf{c}_{i,l}}$ indicates that these LLRs are computed at the relay. In the following, we collect the LLRs for $\mathbf{c}_i$ in the vector $\boldsymbol{L}^R_{\mathbf{c}_i} = (L^R_{\mathsf{c}_{i,1}} \ \cdots \ L^R_{\mathsf{c}_{i,N}})^{\mathrm{T}}$.

Next, the LLRs $\boldsymbol{L}^R_{\mathbf{c}_i}$ are quantized by a scalar quantizer $q_i \colon \mathbb{R} \to \mathcal{Z}_i$ with Q_i quantization levels. We let $\mathcal{Z}_i = \{1, \ldots, Q_i\}$, $i = 1, \ldots, n$, in what follows. The quantizers q_i, $i = 1, \ldots, n$, are matched to the average SNRs $\overline{\gamma}_{i,R}$ of the respective source-relay channels. The quantizer outputs $z_{i,l} = q_i(L^R_{\mathsf{c}_{i,l}})$, $l = 1, \ldots, N$, are collected in the vector $\boldsymbol{z}_i = (z_{i,1} \ \cdots \ z_{i,N})^{\mathrm{T}}$. The vectors $\boldsymbol{z}_i$, $i = 1, \ldots, n$, are then interleaved, yielding $\boldsymbol{z}'_i = \Pi_i(\boldsymbol{z}_i)$. Interleaving is performed to avoid short cycles in the factor graph of the overall network-channel code (cf. Figure 6.5). We note that one of the interleavers Π_i, $i = 1, \ldots, n$, can be omitted. The interleaved quantizer outputs $\boldsymbol{z}'_i$ are jointly encoded by the network encoder yielding $\boldsymbol{z}_R \in \mathcal{Z}_R^M$. The network-coded data $\boldsymbol{z}_R$ is transmitted over the relay-destination channel. The output of the network encoder is matched to the transition probabilities $p(z_D|z_R)$ and the input alphabet $\mathcal{Z}_R$ of the relay-destination channel. A detailed description of the quantization and network coding stages is given in Section 6.4.

We note that in contrast to [108, 109, 118, 119] the relay does not perform soft-output channel decoding which reduces complexity and delay. Moreover, the relay processing does

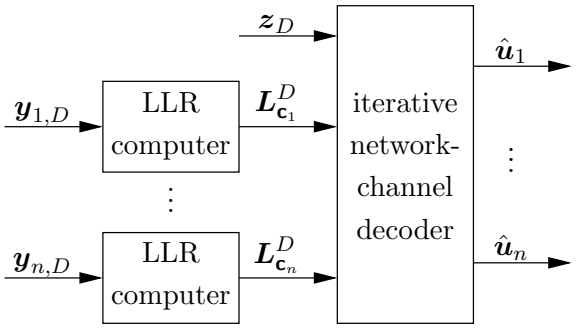


Figure 6.4: Block diagram of the destination.

not depend on the channel codes employed at the sources[1] which increases the flexibility of our scheme. Finally, since all signals are received in consecutive time slots, the processing chains preceding the network encoder shown in Figure 6.3 need to be implemented only once in hardware, thereby reducing chip area.

6.3.3 Destination

Figure 6.4 shows a block diagram of the destination which jointly decodes its received signals. To this end, the destination first computes LLRs for the code bits $\mathbf{c}_i$ as follows:

$$L^D_{\mathsf{c}_{i,l}} = \log \frac{\mathbb{P}\left\{\mathsf{c}_{i,l}=0 \middle| \mathsf{y}^D_{i,\lceil l/m\rceil}=y^D_{i,\lceil l/m\rceil}\right\}}{\mathbb{P}\left\{\mathsf{c}_{i,l}=1 \middle| \mathsf{y}^D_{i,\lceil l/m\rceil}=y^D_{i,\lceil l/m\rceil}\right\}}, \qquad l=1,\ldots,N, \quad i=1,\ldots,n. \tag{6.7}$$

Here, $y^D_{i,\lceil l/m\rceil}$ denotes the receive value corresponding to the symbol which carries the code bit $\mathsf{c}_{i,l}$ and the superscript "D" in $L^D_{\mathsf{c}_{i,l}}$ indicates that these LLRs are computed at the destination. In the following, we collect the LLRs in the vectors $\boldsymbol{L}^D_{\mathbf{c}_i} = (L^D_{\mathsf{c}_{i,1}} \ \cdots \ L^D_{\mathsf{c}_{i,N}})^{\mathrm{T}}$, $i=1,\ldots,n$. The iterative joint network-channel decoder takes $\boldsymbol{z}_D$, i.e., the receive value corresponding to $\boldsymbol{z}_R$, and the LLRs $\boldsymbol{L}^D_{\mathbf{c}_1},\ldots,\boldsymbol{L}^D_{\mathbf{c}_n}$ as inputs and outputs the decoded source data $\hat{\boldsymbol{u}}_1,\ldots,\hat{\boldsymbol{u}}_n$. In Section 6.5, we derive the iterative decoder using a factor graph representation of the overall network-channel code (cf. Figure 6.5). With a particular decoding schedule, this decoder can be viewed as a turbo decoder in which the channel decoders and the network decoder iteratively exchange extrinsic information (cf. Figure 6.6).

6.4 Quantization and Network Encoding

In this section, we discuss the quantization and network encoding stages at the relay in more detail. In both stages, quantization for maximum mutual information is performed (cf. Chapter 5).

[1] To compute the LLRs (6.6), the relay only needs to be aware of the signal constellations employed by the sources.

6.4.1 Quantization

The relay uses one scalar quantizer for each source-relay link to quantize the LLRs $\boldsymbol{L}^R_{\mathsf{c}_i}$. The design of the LLR quantizers is critical for the performance of the system. We aim at maximizing the mutual information $I(\mathsf{c}_{i,l};\mathsf{z}_{i,l})$ between the quantizer output $z_{i,l}$ and the corresponding code bit $c_{i,l}$ for a fixed number of quantization levels Q_i, $i=1,\ldots,n$. To find the quantizer $q_i\colon \mathbb{R}\to\mathcal{Z}_i$, we therefore solve the following optimization problem[2]:

$$q_i = \arg\max_{q} I\big(\mathsf{c}_i; q(L^R_{\mathsf{c}_i})\big), \qquad \text{subject to} \quad |\mathcal{Z}_i| = Q_i. \tag{6.8}$$

It is not hard to see that the problem in (6.8) is equivalent to the quantizer design problem studied in Chapter 5. Therefore, we use Algorithm 5.2 to find an LLR quantizer which solves (6.8). We note that the quantization regions of the optimal quantizers are intervals since we quantize LLRs. The LLR computation (6.6) can be avoided by mapping the quantization intervals for the LLRs to the corresponding quantization regions for the receive values. In this way, quantized LLRs can be obtained directly from the receive values.

We note that quantizer design is performed offline in the proposed scheme. Hence, the relay stores a set of quantizers for a sufficiently wide range of SNRs and then uses for each source-relay channel the quantizer that has been optimized for the corresponding average SNR $\overline{\gamma}_{i,R}$. We have observed that choosing $Q_i=8$ is usually sufficient, i.e., increasing the number of quantization levels beyond 8 yields at best negligible performance gains (which is in line with our findings in Section 5.6).

6.4.2 Network Encoding

In the following, we let $\boldsymbol{Z}=(\boldsymbol{z}'_1\ \cdots\ \boldsymbol{z}'_n)$ and $\boldsymbol{C}=(\boldsymbol{c}'_1\ \cdots\ \boldsymbol{c}'_n)$ be matrices whose ith columns are respectively the interleaved quantizer outputs and the corresponding code bits transmitted by the ith source, $i=1,\ldots,n$ (note that $\boldsymbol{c}'_i=\Pi_i(\boldsymbol{c}_i)$). We denote the rows of the $N\times n$ matrices $\boldsymbol{Z}$ and $\boldsymbol{C}$ by $\boldsymbol{r}_j^{\mathrm{T}}$ and $\boldsymbol{d}_j^{\mathrm{T}}$, $j=1,\ldots,N$, i.e., we have $\boldsymbol{Z}=(\boldsymbol{r}_1\ \cdots\ \boldsymbol{r}_N)^{\mathrm{T}}$ and $\boldsymbol{C}=(\boldsymbol{d}_1\ \cdots\ \boldsymbol{d}_N)^{\mathrm{T}}$. Furthermore, we denote by $\boldsymbol{a}_{\sim j}$ the vector obtained by removing the jth element from the vector $\boldsymbol{a}$.

The network encoder at the relay maps M of the N rows of $\boldsymbol{Z}$ to an element of $\mathcal{Z}_R$, i.e., the input alphabet of the relay-destination channel (recall that we have assumed M channel uses for the relay). Hence, the network encoder is a function $g\colon \mathcal{Z}_1\times\cdots\times\mathcal{Z}_n\to\mathcal{Z}_R$, with $z_{R,k}=g(\boldsymbol{r}_{j_k})$, $k=1,\ldots,M$. Here, $j_k\in\{1,\ldots,N\}$ denotes the index of the row of $\boldsymbol{Z}$ which is mapped to the kth output of the network encoder. The transmit signal of the relay is given by $\boldsymbol{z}_R=(z_{R,1}\ \cdots\ z_{R,M})^{\mathrm{T}}$ and the corresponding receive signal at the destination, $\boldsymbol{z}_D$, is the output of a DMC with transition probabilities $p(\boldsymbol{z}_D|\boldsymbol{z}_R)=\prod_{k=1}^{M}p(z_{D,k}|z_{R,k})$.

The design of the network coding function g is motivated by the iterative decoding procedure at the destination (see Section 6.5 for details). To maximize the information transfer

[2]For the sake of notational clarity, we suppress the bit position index l in what follows.

between the individual channel decoders, we seek to maximize $I(\mathsf{d}_{j_k,i};\mathsf{z}_{D,k}|\mathbf{d}_{j_k,\sim i})$ for each $i \in \{1,\ldots,n\}$ (here, $d_{j_k,i}$ denotes the ith element of the vector $\boldsymbol{d}_{j_k}$). Loosely speaking, given perfect *a priori* information $\boldsymbol{d}_{j_k,\sim i}$, $z_{D,k}$ should be as informative about $d_{j_k,i}$ as possible. However, since $I(\mathsf{d}_{j_k,i};\mathsf{z}_{D,k}|\mathbf{d}_{j_k,\sim i})$ cannot be maximized for each i independently, we resort to maximizing a function of these mutual information expressions. Extending the case $n = 2$ (cf. [108]), we propose to maximize the following objective function[3]:

$$\frac{1}{n}\sum_{i=1}^{n} I(\mathsf{d}_i;\mathsf{z}_D|\mathbf{d}_{\sim i}) = I(\mathbf{d};\mathsf{z}_D) - \frac{1}{n}\sum_{i=1}^{n} I(\mathbf{d}_{\sim i};\mathsf{z}_D) \tag{6.9}$$

$$= I(\mathbf{d};\mathbf{r}) - I(\mathbf{d};\mathbf{r}|\mathsf{z}_D) - \frac{1}{n}\sum_{i=1}^{n}\Big(I(\mathbf{d}_{\sim i};\mathbf{r}_{\sim i}) - I(\mathbf{d}_{\sim i};\mathbf{r}_{\sim i}|\mathsf{z}_D)\Big). \tag{6.10}$$

In (6.9) and (6.10), we have used the chain rule of mutual information and the fact that $\mathbf{d} \leftrightarrow \mathbf{r} \leftrightarrow g(\mathbf{r}) \leftrightarrow \mathsf{z}_D$ forms a Markov chain. With (6.10), the optimal network encoder can be written as (note that the terms $I(\mathbf{d};\mathbf{r})$ and $I(\mathbf{d}_{\sim i};\mathbf{r}_{\sim i})$ in (6.10) do not depend on g)

$$g^\star = \arg\min_g I(\mathbf{d};\mathbf{r}|\mathsf{z}_D) - \frac{1}{n}\sum_{i=1}^{n} I(\mathbf{d}_{\sim i};\mathbf{r}_{\sim i}|\mathsf{z}_D). \tag{6.11}$$

Writing mutual information in terms of relative entropy, we can reformulate (6.11) as

$$g^\star = \arg\min_g \mathbb{E}\big\{D\big(p(\boldsymbol{d}|\mathbf{r})\|p(\boldsymbol{d}|\mathsf{z}_D)\big)\big\} - \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\big\{D\big(p(\boldsymbol{d}_{\sim i}|\mathbf{r}_{\sim i})\|p(\boldsymbol{d}_{\sim i}|\mathsf{z}_D)\big)\big\}. \tag{6.12}$$

We note that the optimization problem (6.12) is similar to the channel-optimized vector quantizer design discussed in Section 5.5. In fact, the proposed network encoder performs channel-optimized vector quantization with a modified objective function compared to Section 5.5. We can therefore solve (6.12) using Algorithm 5.4. To this end, we replace the variables $\mathbf{x} \leftrightarrow \mathbf{y} \leftrightarrow \mathsf{z} \leftrightarrow \tilde{\mathsf{z}}$ of Algorithm 5.4 by $\mathbf{d} \leftrightarrow \mathbf{r} \leftrightarrow \mathsf{z}_R \leftrightarrow \mathsf{z}_D$ and we use the modified cost function

$$C(\boldsymbol{r}, z_D) = D\big(p(\boldsymbol{d}|\boldsymbol{r})\|p(\boldsymbol{d}|z_D)\big) - \frac{1}{n}\sum_{i=1}^{n} D\big(p(\boldsymbol{d}_{\sim i}|\boldsymbol{r}_{\sim i})\|p(\boldsymbol{d}_{\sim i}|z_D)\big) \tag{6.13}$$

in line number 9 of Algorithm 5.4.

We note that the proposed network coding strategy is fundamentally different from conventional algebraic network coding [2]. However, the proposed coding strategy at the relay combines the data of different sources which justifies the use of the term *network coding*. The choice of the objective function in (6.9) ensures that the data of the individual sources has large (little) impact on the network-coded data if the respective source-relay SNR is high (low). This allows the proposed scheme to perform well in asymmetric channel conditions (cf. [106]) since it prevents sources with poor SNR from rendering the network-coded data useless.

[3]For the sake of notational simplicity, we suppress the indices k and j_k in what follows.

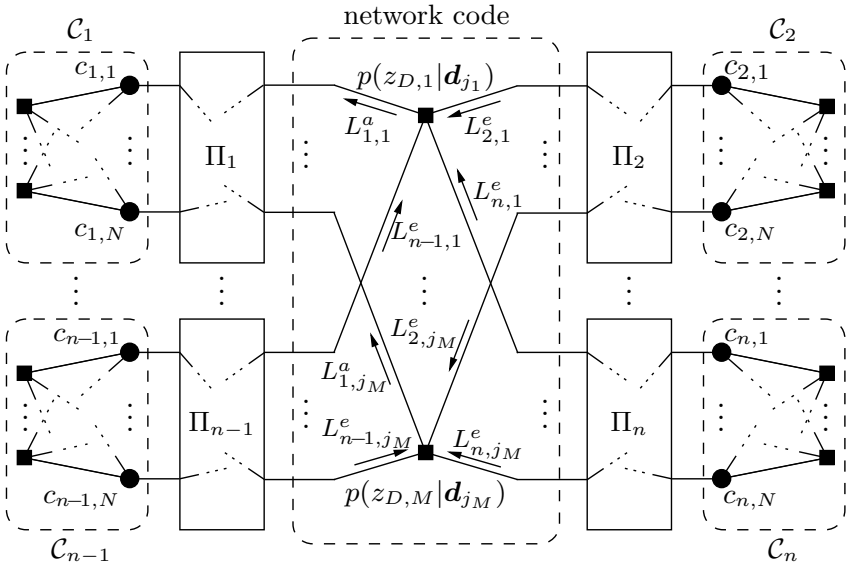


Figure 6.5: Factor graph of the overall network-channel code.

The joint distribution $p(\boldsymbol{d},\boldsymbol{r}) = \prod_{i=1}^{n} p(r_i|d_i)p(d_i)$ is required for the design of the network encoder. We note that $p(r_i|d_i)$ is obtained from the design of the scalar LLR quantizers and $p(d_i)$ (i.e., the prior distribution of the code bits) is known *a priori*. The network encoder $g\colon \mathcal{Z}_1\times\cdots\times\mathcal{Z}_n \to \mathcal{Z}_R$ can be implemented using an n-dimensional lookup table that is indexed by the quantizer outputs $\boldsymbol{r}$. Furthermore, the network encoder can be designed on-the-fly during data transmission since $p(\boldsymbol{d},\boldsymbol{r})$ is known once the relay has chosen the LLR quantizers. Hence, it is not necessary to optimize the network encoder in advance for sufficiently many combinations of source-relay SNRs. Finally, we note that the LLR quantizers and the network encoder may depend on the bit position if the sources use higher order signal constellations.

6.5 Iterative Joint Network-Channel Decoder

The processing at the relay creates an equivalent DMC with input $\boldsymbol{d}_{j_k}$, output $z_{D,k}$, and transition probability mass function (pmf) $p(z_{D,k}|\boldsymbol{d}_{j_k}) = p(z_{D,k}|c'_{1,j_k},\ldots,c'_{n,j_k})$, $k = 1,\ldots,M$. The pmf $p(z_{D,k}|\boldsymbol{d}_{j_k})$ corresponds to the network code in our scheme. It couples the code bits of all sources and thus enables joint decoding of the source codewords. We note that $p(z_{D,k}|\boldsymbol{d}_{j_k})$ is known once the network encoder at the relay is fixed[4]. The operation of the proposed joint network-channel decoder is such that the individual channel decoders iteratively exchange extrinsic LLRs via the network decoder (6.14). Figure 6.5 shows the factor graph of the overall network-channel code.

The messages that are exchanged between the channel decoders and the network decoder are the extrinsic LLRs L^e_{i,j_k} and the prior LLRs L^a_{i,j_k}, $i = 1,\ldots,n$, $k = 1,\ldots,M$. The

[4]Strategies for making the pmf $p(z_{D,k}|\boldsymbol{d}_{j_k})$ available at the destination are discussed at the end of this section.

network decoder computes the prior LLR L^a_{i,j_k} for the ith channel decoder using the $n-1$ extrinsic LLRs L^e_{l,j_k} $(l \neq i)$ and the local function $p(z_{D,k}|\boldsymbol{d}_{j_k})$, where the $z_{D,k}$ is the receive value corresponding to $z_{R,k}$. The sum-product update rule (cf. Section 2.5) for the prior LLRs L^a_{i,j_k}, $k = 1, \ldots, M$ is given by

$$L^a_{i,j_k} = \log \frac{\mu_{p \to c'_{i,j_k}}(c'_{i,j_k}=0)}{\mu_{p \to c'_{i,j_k}}(c'_{i,j_k}=1)}, \qquad i = 1, \ldots, n, \tag{6.14a}$$

where

$$\mu_{p \to c'_{i,j_k}}(c'_{i,j_k}) = \sum_{c'_{1,j_k}} \cdots \sum_{c'_{i-1,j_k}} \sum_{c'_{i+1,j_k}} \cdots \sum_{c'_{n,j_k}} p(z_{D,k}|c'_{1,j_k}, \ldots, c'_{n,j_k}) \prod_{l:l\neq i} \mu_{c'_{l,j_k} \to p}(c'_{l,j_k}), \tag{6.14b}$$

and

$$\mu_{c'_{l,j_k} \to p}(c'_{l,j_k}=b) = \frac{\exp\big(-bL^e_{l,j_k}\big)}{1+\exp\big(-L^e_{l,j_k}\big)}, \qquad b \in \{0,1\}. \tag{6.14c}$$

A multitude of message passing schedules may be used with the proposed joint network-channel decoder, the most common being *flooding* and *serial* schedules. In the flooding schedule, *all* channel decoders update the extrinsic LLRs and in the next step the network decoder updates *all* prior LLRs. In contrast, for the serial schedule only *one* channel decoder updates its extrinsic LLRs and then the network decoder updates the prior LLRs only for the *next scheduled* channel decoder. The complexity per iteration of the joint network-channel decoder is the same for both schedules. However, we found that the serial schedule outperforms the flooding schedule in terms of convergence speed. Therefore, we consider only the serial schedule described above which corresponds to the turbo-like joint network-channel decoder depicted in Figure 6.6. Note that in contrast to a turbo decoder, the proposed decoder exchanges extrinsic LLRs and prior LLRs for the *code* bits.

Finally, we mention two strategies for making the pmf $p(z_{D,k}|\boldsymbol{d}_{j_k})$ available to the destination. One possibility is to communicate $p(z_{D,k}|\boldsymbol{d}_{j_k})$ directly from the relay to the destination. However, this approach leads to high communication overhead since $2^n|\mathcal{Z}_D|$ probabilities have to be transmitted with sufficiently high accuracy. The second strategy is to store the set of scalar quantizers employed by the relay also at the destination and communicate only the quantizer choice at the relay (i.e., n integer-valued indices). Then, the destination designs the network encoder in the same manner as the relay. The pmf $p(z_{D,k}|\boldsymbol{d}_{j_k})$ is a by-product of the network encoder optimization. This strategy is clearly preferable if computational overhead at the destination is cheaper than communication overhead on the relay-destination link. In any case, we assume that $p(z_{D,k}|\boldsymbol{d}_{j_k})$ is available at the destination and neglect the signaling overhead.

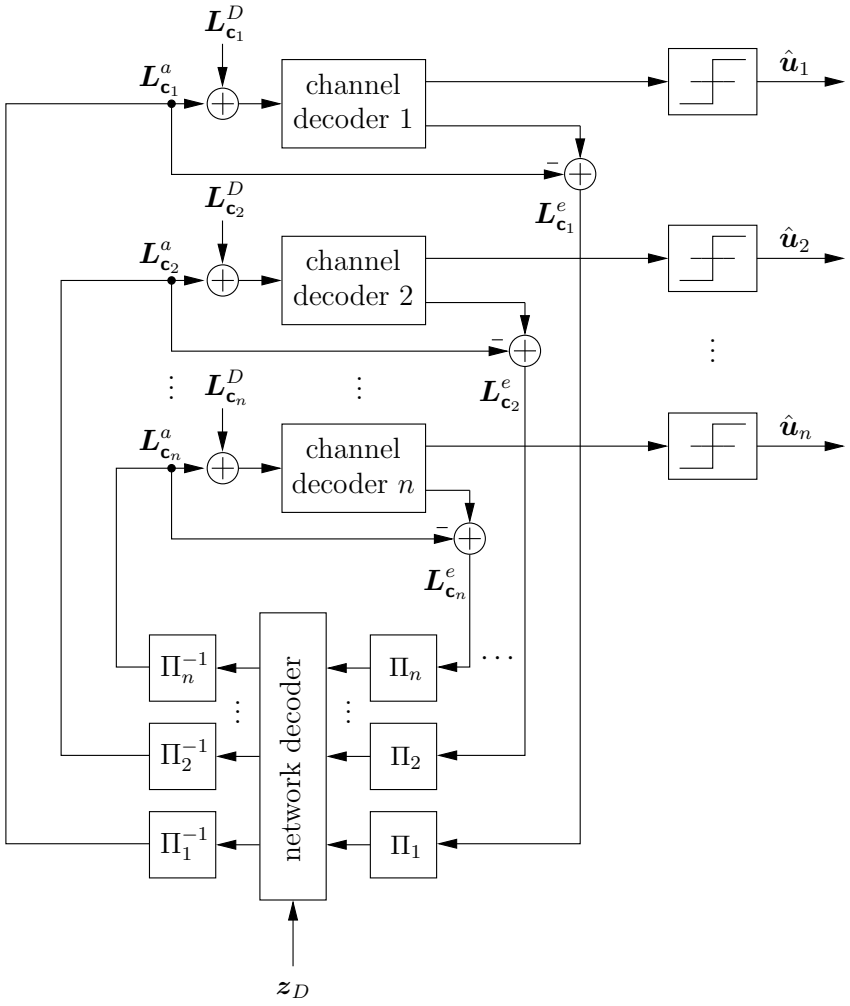


Figure 6.6: Iterative turbo-like joint network-channel decoder.

6.6 Numerical Results

In this section we assess the performance of the proposed transmission scheme by Monte Carlo simulations. We analyze the bit error rate (BER) and the frame error rate (FER) for the transmission over AWGN channels and block-fading channels, respectively.

6.6.1 General Setup

Each source generates $K = 2^{10}$ independent and equally likely information bits. A recursive convolutional code with generator polynomials $(1, 13/15)_8$ is used to encode the data. The output of the channel encoder is punctured to achieve a code rate of $R = (n+1)/(2n)$. A binary phase-shift keying signal constellation ($m = 1$) is used to transmit the code bits and therefore each source uses the channel $M = K/R = 2nK/(n+1)$ times. We note that the above choice for R yields $R_s = 1/2$ and we have $R \to 1/2$ as $n \to \infty$. We set $E_s = 4K/3$ and, hence, $P_s = E_s/M = 2(n+1)/(3n)$, which corresponds to $P_s = 1$ for $n = 2$ sources. Note that $N = M$ (since $m = 1$) and thus the relay encodes all rows of the matrix $\boldsymbol{Z}$ (cf. Subsection 6.4.2), i.e., $\boldsymbol{d}_{j_k} = \boldsymbol{d}_k$, $k = 1, \ldots, M$.

We assume that the source-relay channels and the source-destination channels are symmetric, i.e., we have $d_R \triangleq d_{i,R}$, $d_D \triangleq d_{i,D}$ and thus $\overline{\gamma}_R \triangleq \overline{\gamma}_{i,R}$, $\overline{\gamma}_D \triangleq \overline{\gamma}_{i,D}$. The path-loss exponent in (6.1) is chosen as $\alpha = 3.52$. We model the relay-destination channel as 3 parallel binary symmetric channels with bit error probability P_b. Hence, we have $\mathcal{Z}_R = \mathcal{Z}_D = \{0, 1, \ldots, 7\}$ and the transition probabilities are given as

$$p(z_D|z_R) = P_b^{d_H(\boldsymbol{b}_{z_D}, \boldsymbol{b}_{z_R})}(1-P_b)^{3-d_H(\boldsymbol{b}_{z_D}, \boldsymbol{b}_{z_R})}, \tag{6.15}$$

where $d_H(\cdot,\cdot)$ denotes the Hamming distance and $\boldsymbol{b}_{z_R}$, $\boldsymbol{b}_{z_D}$ are the binary labels corresponding to z_R and z_D, respectively. The LLR quantizers at the relay use 8 quantization levels. The joint network-channel decoder at the destination performs 5 iterations and uses a serial schedule. We have used random interleavers with depths equal to the block length.

In our setting, the code rate R decreases as n increases and thus the blocklength increases with n. At the same time, the transmit power decreases with increasing n since the total transmit energy of each source is fixed to E_s. Moreover, the available rate on the relay-destination channel has to be shared between more sources as n grows. The changes in the system parameters for varying n are summarized in Table 6.1. We observe that the rate loss due to the half-duplex relay node can be reduced by a factor of 2/3 when 6 sources share the relay instead of only 2 sources.

6.6.2 Constant Channels

We first study the performance of the proposed scheme when the source-relay and source-destination channels are constant. In particular, we let $\mathsf{h}_{i,j} = 1$ and $d_R = 0.6754 \cdot d_D$ in (6.1), i.e., we consider AWGN channels. In terms of the source-relay and source-destination

Table 6.1: Changes in the system parameters for $n = 2, \ldots, 7$. Percentage changes relative to $n = 2$ are given in parentheses.

n	R	ΔR_s	P_s
2	0.750	0.250	1.000
3	0.667 (-11.1 %)	0.167 (-33.3 %)	0.889 (-11.1 %)
4	0.625 (-16.7 %)	0.125 (-50.0 %)	0.833 (-16.7 %)
5	0.600 (-20.0 %)	0.100 (-60.0 %)	0.800 (-20.0 %)
6	0.583 (-22.2 %)	0.083 (-66.7 %)	0.778 (-22.2 %)
7	0.571 (-23.8 %)	0.071 (-71.4 %)	0.762 (-23.8 %)

SNRs this corresponds to $[\gamma_R]_{\mathrm{dB}} = [\gamma_D]_{\mathrm{dB}} + 6\,\mathrm{dB}$. Moreover, we assume an error-free relay-destination channel, i.e., we have $P_b = 0$. For this setting, Figure 6.7 shows the BER performance for 2 to 7 sources. We also plot the performance without relay (at the same sum rate) and the performance of a scheme which decodes at the relay and forwards the modulo-2 sums $\bigoplus_{l=1}^{n} d_{k,l}$, $k = 1, \ldots, M$, in case of successful decoding.

We observe that the performance is significantly improved for BER values of interest when more sources share the relay. The SNR gain saturates as the number of sources increases (diminishing returns). At a BER of 10^{-4} the system with 7 sources gains 1.3 dB over the system with 2 sources. We note that this behavior is observed for a wide range of sum rates and geometries. Furthermore, the proposed scheme clearly outperforms the "XOR" scheme described above, irrespective of the number of sources. Compared to a transmission without relay, substantial SNR gains of more than 3 dB are obtained.

In Figure 6.8, we study the influence of P_b on the BER for the case of 2 sources. We compare the network encoders which take the noisy relay-destination channel into account (solid lines) to network encoders which falsely assume $P_b = 0$ (dashed lines). The channel-optimized network encoder design yields a graceful performance degradation as P_b increases and improves substantially over the non-channel-optimized design. Assuming $P_b = 0$ when in fact $P_b > 0$ quickly deteriorates the BER performance. For P_b as small as 10^{-2}, the BER saturates at about 10^{-3} if the relay falsely assumes $P_b = 0$.

Figure 6.9 shows the influence of P_b for the case of 6 sources. In this case, the difference between the channel-optimized design and the non-channel-optimized design is still significant but less pronounced. Specifically, the BER saturates also when the correct values of P_b is taken into account in the network encoder design. However, e.g., for $P_b = 0.1$, the channel-optimized design improves by more than one order of magnitude in terms of BER over the network encoders which incorrectly assume $P_b = 0$.

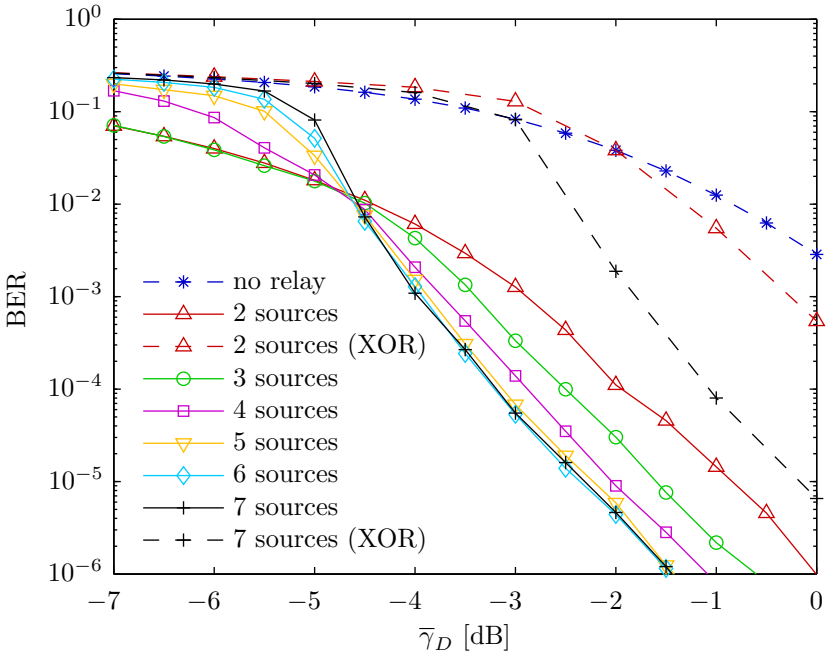


Figure 6.7: BER performance in the AWGN case for 2 to 7 sources with $P_b = 0$.

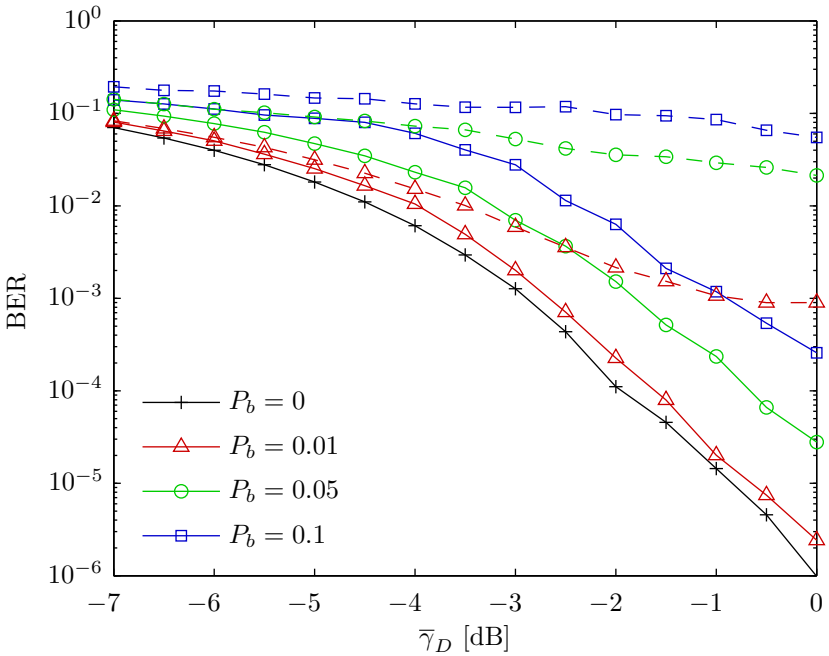


Figure 6.8: Dependence of the BER on P_b in the AWGN case for 2 sources. Solid lines correspond to the proposed network encoder design and dashed lines correspond to network encoders which falsely assume $P_b = 0$.

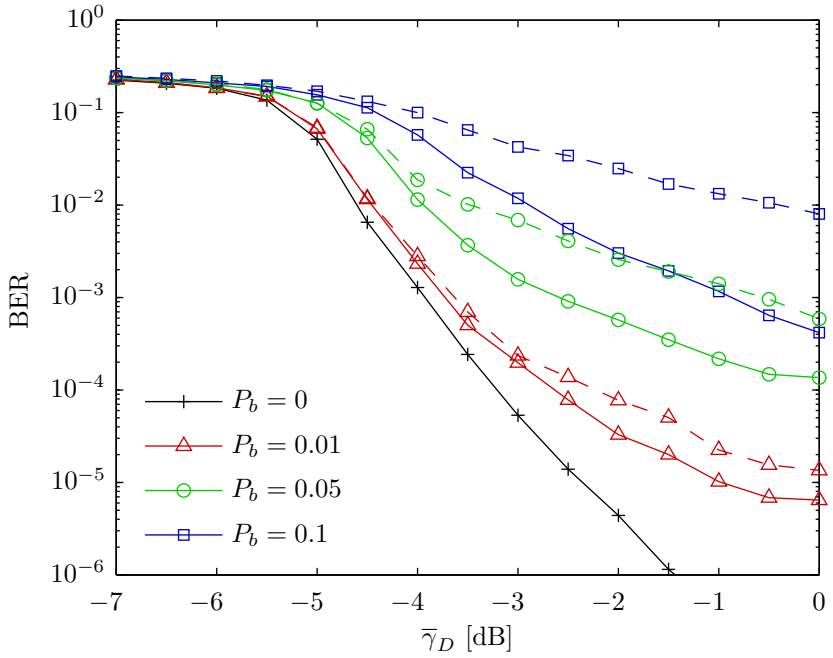


Figure 6.9: Dependence of the BER on P_b in the AWGN case for 6 sources. Solid lines correspond to the proposed network encoder design and dashed lines correspond to network encoders which falsely assume $P_b = 0$.

6.6.3 Block-Fading Channels

We next consider the case where the source-relay and source-destination channels are block-fading channels with $\mathsf{h}_{i,j} \sim \mathcal{CN}(0, 1)$. The geometry of the MARC is chosen such that the average source-relay and source-destination SNRs are related as $[\overline{\gamma}_R]_{\mathrm{dB}} = [\overline{\gamma}_D]_{\mathrm{dB}} + 3\,\mathrm{dB}$. Figure 6.10 shows the FER performance of the proposed scheme for 2 and 7 sources when $P_b = 0$. We again use a transmission without relay (at the same sum rate) and the "XOR" coding strategy at the relay as baselines.

We observe that the FER performance degrades only very slightly (by approximately 0.5 dB) when the number of sources is increased from 2 to 7. Moreover, the proposed scheme simultaneously provides second-order diversity for *all* sources. An SNR gain of more than 8 dB compared to a transmission without relay is achieved for 7 sources at an FER of 10^{-2}. Furthermore, the "XOR" transmission scheme is outperformed by about 3 dB.

In Figure 6.11, we study the influence of P_b on the FER for the case of 2 sources. We again observe a graceful performance degradation as P_b increases. The channel-optimized network encoder outperforms its non-channel-optimized counterpart by about 1 dB for the values of P_b shown in 6.11. It is important to note that the diversity order does not decrease as P_b increases.

Figure 6.12 shows the dependence of the FER on P_b for the case of 6 sources. Compared to the case of 2 sources, the gap in terms of SNR between the channel-optimized network

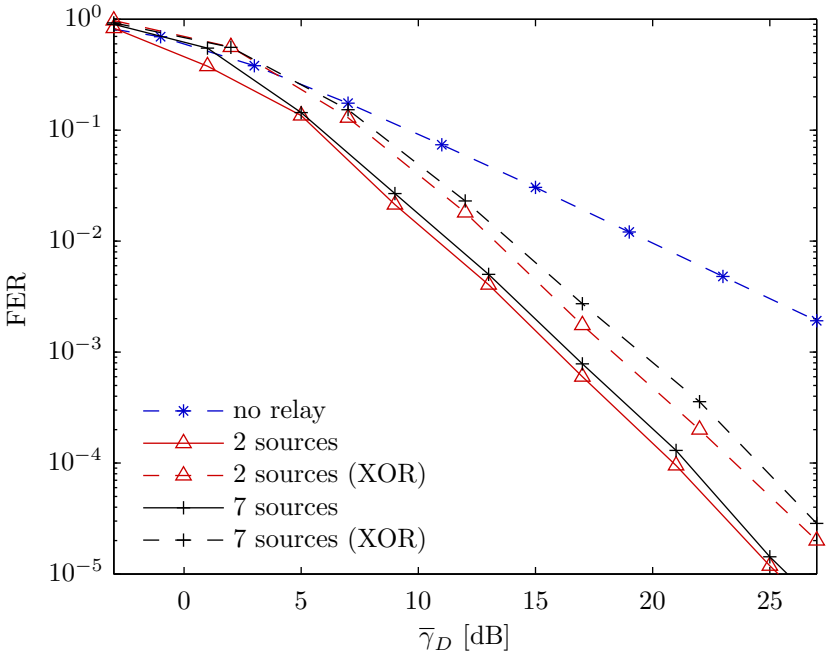


Figure 6.10: FER performance in the block-fading case for 2 and 7 sources with $P_b = 0$.

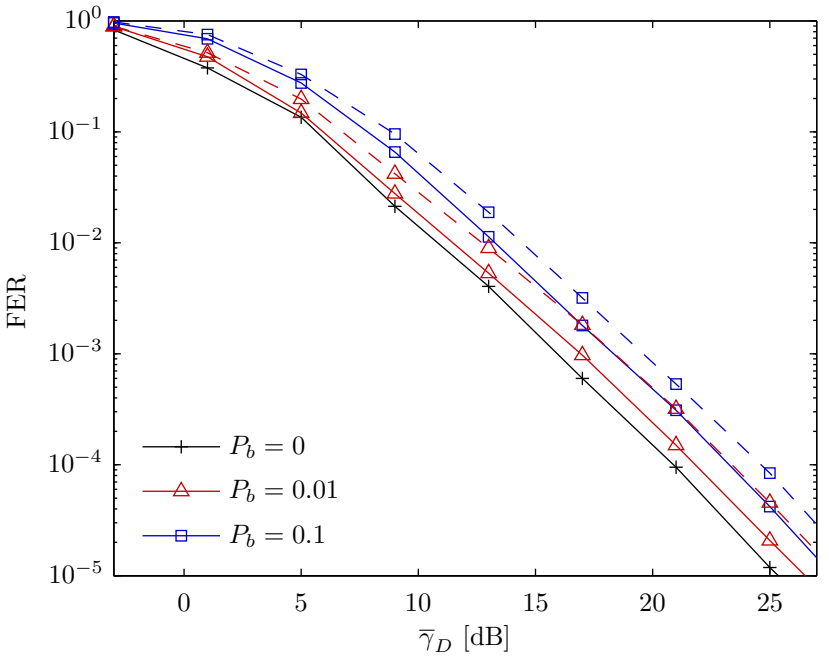


Figure 6.11: Dependence of the FER on P_b in the block-fading case for 2 sources. Solid lines correspond to the proposed network encoder design and dashed lines correspond to network encoders which falsely assume $P_b = 0$.

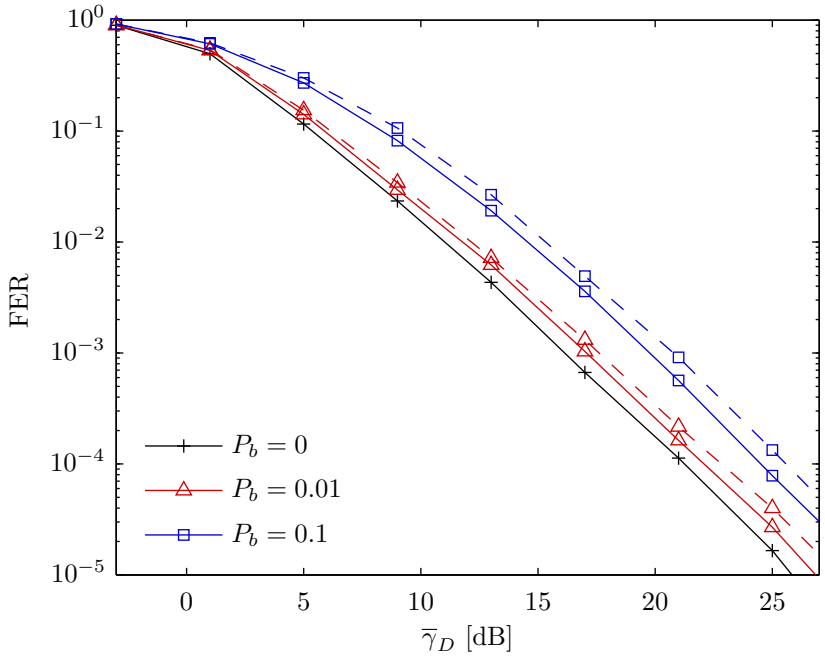


Figure 6.12: Dependence of the FER on P_b in the block-fading case for 6 sources. Solid lines correspond to the proposed network encoder design and dashed lines correspond to network encoders which falsely assume $P_b = 0$.

encoder and the non-channel-optimized network encoder is reduced. On the other hand, the performance deteriorates more quickly as P_b increases. These observations are in accordance with the observations made in the AWGN case. We note that the proposed scheme provides a diversity order of two for all 6 sources also when the relay-destination channel is error-prone.

6.6.4 Blind Performance Estimation

Finally, we apply the blind performance estimators proposed in Chapter 3 and compare their results to the conventional unbiased nonblind BER and FER estimators. In Figure 6.13, we compare blind BER estimation to nonblind BER estimation in the AWGN case with 2 and 7 sources. We observe that for lower BER values, the blind estimator underestimates the BER and is about 0.5 dB away in terms of SNR from the result of nonblind estimator. It seems that the bias of the blind estimator does not significantly depend on the number of sources. Figure 6.14 shows the result of blind FER estimation in the block-fading case with 2 and 7 sources. In this case, the blind estimate differs only very slightly (0.1 dB to 0.2 dB) from the unbiased nonblind estimate. It is important to recall that the blind FER estimator is always biased when coding is used since it assumes independence of the bit errors.

Although the blind estimators are biased (which is expected since we perform suboptimal iterative decoding), our results show that they are useful also in this cooperative communications setting. The blind FER estimate is almost equal to the nonblind FER estimate.

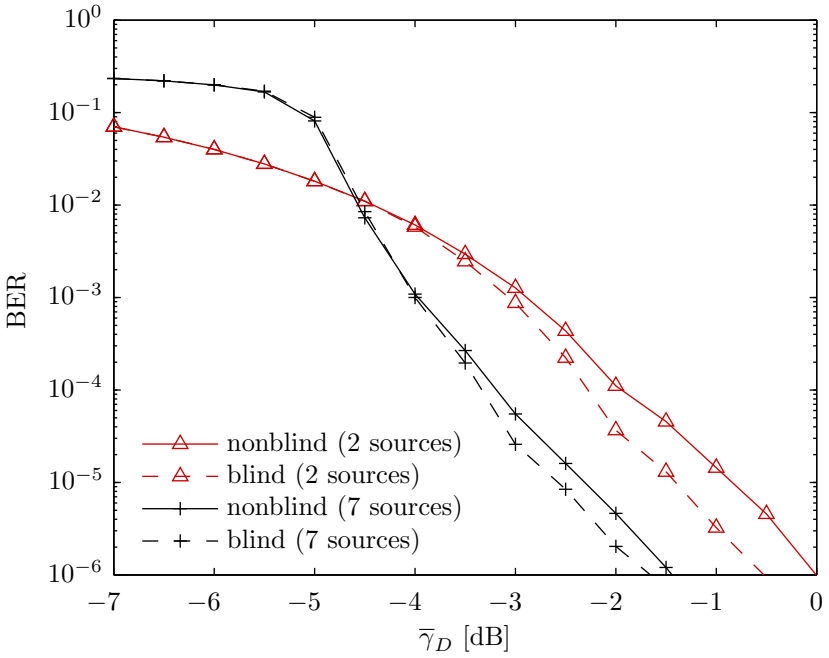


Figure 6.13: Blind BER estimation (dashed lines) in the AWGN case compared to nonblind BER estimation for 2 and 7 sources.

The blind BER estimate differs from the nonblind BER estimate when the error floor of the turbo-like decoder comes into play. This behavior is also observed with regular turbo codes (cf. Figure 3.23).

6.7 Discussion

In this chapter, we have studied relay-based cooperative communication for the MARC with two or more sources and network coding at the physical layer. The proposed transmission scheme scales well with the number of sources and is simple to implement. In particular, the relay performs scalar quantization followed by a network encoding operation which can be implemented using a lookup table. The design of the quantizers and the network encoder is performed using algorithms that we have presented in Chapter 5. In contrast to other transmission schemes for the MARC, the relay does not perform (soft-output) channel decoding, thereby reducing computational complexity and delay. The destination uses an iterative network-channel decoder to jointly decode the source data. We have derived this decoder using the update rules of the sum-product algorithm on the factor graph representation of the overall network-channel code. Our numerical results confirm the excellent performance of the proposed transmission scheme. The channel-optimized design of the network encoder effectively combats the noise on the relay-destination channel. In the case of block-fading channels, a diversity order of two is achieved for all sources simultaneously. Furthermore,

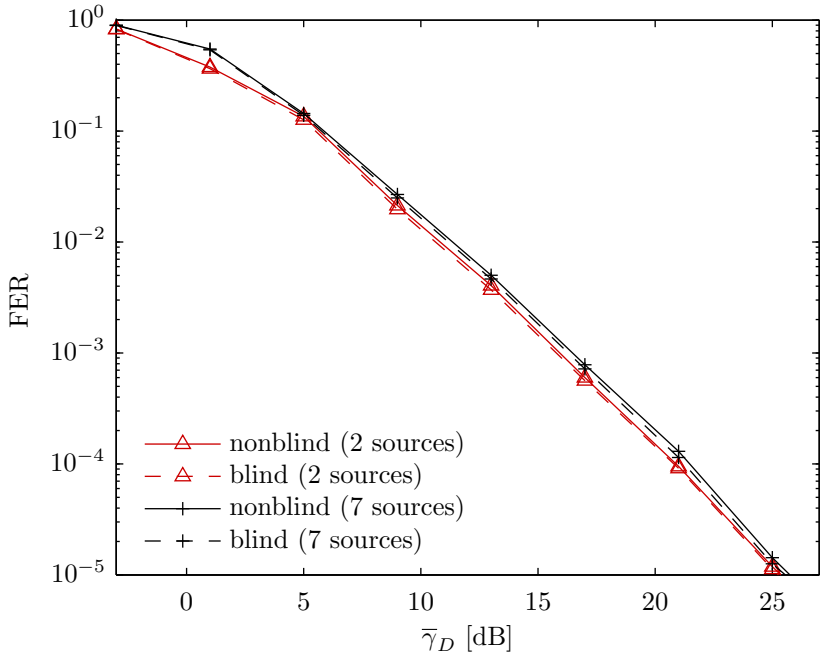


Figure 6.14: Blind FER estimation (dashed lines) in the block-fading case compared to nonblind FER estimation for 2 and 7 sources.

we have presented simulation results which underpin the usefulness of the blind performance estimators presented in Chapter 3.

There are numerous possible extensions of the proposed transmission scheme which, however, are outside the scope of this thesis. An extension to correlated source data is useful, e.g., in sensor networks where correlated measurements have to be transmitted to other network nodes. The network code could be optimized subject to constraints on the degrees of the corresponding factor nodes. This allows us to limit the decoding complexity even for a large number of sources. Moreover, the relay could adaptively choose between different network coding strategies depending on the channel conditions. For example, if the relay-destination channel does not form a bottleneck, then it may be beneficial to forward data without performing network coding (cf. [105]).

7

Conclusions

In this concluding chapter, we give a concise summary of the main contributions of this thesis (cf. Section 7.1). In addition, we outline open problems beyond the results presented in this thesis that may provide directions for further research (cf. Section 7.2).

7.1 Summary of Contributions

Blind Performance Estimation for Bayesian Detectors

- We proposed unbiased and consistent blind estimators for the (conditional) error probabilities, the minimum mean-square error (MSE), and the mutual information of Bayesian detectors. We proved that the blind estimator for the unconditional error probability always dominates the corresponding nonblind estimator in terms of MSE. For the conditional error probabilities, we gave conditions under which the blind estimators outperform the corresponding nonblind estimators for arbitrary distributions of the data.

- The proposed blind estimators are based on soft information, i.e., on the posterior probabilities of the hypotheses or on the log-likelihood ratio (LLR) in the binary case. Our results show that in almost all cases soft information improves the accuracy of performance estimation. We therefore conclude that if soft information is available, then it should be used for performance estimation.

- Our blind estimators compute the sample mean of functions of the *a posteriori* probabilities (APPs) and are therefore simple to implement. Hence, Bayesian detectors can compute their performance online as a by-product with little to no extra cost. This may be useful for a variety of adaptive systems, e.g., systems employing adaptive modulation and coding in the communications context.

- We derived the Cramér-Rao lower bound for bit error probability estimation with conditionally Gaussian LLRs under maximum *a posteriori* detection. We showed that in this case an efficient estimator does not exist.

- We studied the properties of LLRs and we presented novel relations between the conditional and unconditional moments of functions of LLRs. These relations are based on the so called "consistency property" that connects the conditional and the unconditional LLR distributions such that any one of the three distributions suffices to express the other two.

- We presented application examples for the proposed blind estimators. Our numerical results confirm the usefulness of the blind estimators even in cases with model uncertainty and approximate LLR computation.

The Rate-Information Trade-off in the Gaussian Case

- We derived closed-form expressions for the information-rate function and the rate-information function. Furthermore, we showed that MSE-optimal (noisy) source coding is suboptimal in terms of the rate-information trade-off.

- In the vector case, the rate-information trade-off is determined by the allocation of the compression rate to the individual modes. The optimal rate-information trade-off is achieved by performing reverse waterfilling on the mode signal-to-noise ratios. Hence, suitable linear filtering with subsequent MSE-optimal source coding is sufficient to achieve the optimal rate-information trade-off.

- Our results show that the Gaussian information bottleneck (GIB) is equivalent to linear filtering with subsequent MSE-optimal compression. We thus established a connection between rate-distortion (RD) theory and the GIB. Therefore, the RD theorem provides achievability and converse for the rate-information trade-off in the Gaussian case. This entails that the information-rate function is indeed the dividing line between the achievable and not achievable rate-information pairs.

- We designed scalar quantizers and we compared their performance to the optimal rate-information trade-off. It turned out that the information-rate function can be closely approached by MSE-optimal scalar quantizers.

Quantizer Design for Communication Problems

- We presented algorithms for mutual-information-optimal quantizer design. For scalar quantizer design we derived an algorithm which performs an alternating optimization and is reminiscent of the Lloyd-Max algorithm. While this algorithm is simple to implement, we also proposed a greedy algorithm which is even simpler as it avoids root-finding. Furthermore, we found that the maximum output entropy quantizer is a suitable initialization for our algorithms.

- We extended the concept of channel-optimized vector quantization to mutual information as optimality criterion. To solve the corresponding optimization problem we proposed an algorithm which is based on the iterative information bottleneck (IB) algorithm. A major advantage of our algorithm is that it finds optimized labels for the quantizer output and thus avoids the NP-hard label optimization problem.

- We compared the performance of mutual-information-optimal LLR quantizers to the optimal rate-information trade-off. It turned out that scalar quantizers approach the information-theoretic limit very closely. This implies that vector quantization (VQ) of independent LLRs may at best yield a negligible performance improvement.

Quantization-Based Network Coding for the MARC

- We presented a cooperative transmission scheme for the multiple-access relay channel (MARC) with two or more sources, which performs network coding at the physical layer. In our scheme, the relay essentially performs LLR quantization followed by a network coding operation that can be implemented using a table lookup. Hence, the proposed

scheme allows for a low-complexity implementation of the relay which is important in practice.

- We designed the network encoder at the relay as a channel-optimized vector quantizer. Hence, the network encoder takes the noise on the relay-destination channel into account. We found that it is important to consider channel-optimized network encoders since otherwise a significant performance penalty is incurred.

- We used a factor graph approach to derive the joint network-channel decoder that is used at the destination. We analyzed the performance of this iterative decoder using Monte Carlo simulations and practical channel codes. It turned out that the proposed scheme yields significant coding gains compared to baseline schemes. Furthermore, for block fading channels a diversity order of two is achieved for all sources simultaneously.

7.2 Open Problems

Blind Performance Estimation for Bayesian Detectors

- The proposed blind estimators are unbiased if (a) there is no uncertainty in the data model and (b) the APPs are computed exactly. In many applications these conditions are not satisfied, e.g., due to errors in the estimation of the data model and due to limited computational resources which necessitate approximations.

  Our numerical results show that the blind estimators produce useful results even if the above conditions are not met, i.e., their bias is acceptably small in the considered cases. However, a numerical case-by-case analysis is rather cumbersome. Instead, it would be desirable to have upper bounds on the bias that depend on the amount of mismatch in the data model and the APP computation.

- In the binary case, LLR correction can be used to eliminate the bias. Unfortunately, LLR correction cannot be performed in a blind manner. However, (3.33) seems to offer a way to perform a blind "consistency check" of the LLRs since (3.33) is based on the consistency property. Replacing the expectation on the right-hand side of (3.33) by a sample mean and computing the difference to the prior probability may be a suitable measure for how inconsistent approximate LLRs are. Having such a measure is an important step towards approximate blind LLR correction.

The Rate-Information Trade-off in the Gaussian Case

- We considered the optimization of the compression mapping for the case where the data and the relevance variable are jointly Gaussian. In some situations it may be of interest to jointly optimize the compression and the distribution of the relevance variable (cf. (4.72)). In the communications context this joint optimization gives rise to the capacity

of a quantized channel. Unfortunately, it is unclear how to solve this harder problem since the variables are coupled in an intricate way.

- The rate-information trade-off is known in closed form only in very few cases, which is similar to the RD trade-off. It would be desirable to find additional cases which allow for closed-form expressions of the optimal rate-information trade-off. A case of particular interest in many communication problems is the Gaussian channel with binary input.

Quantizer Design for Communication Problems

- As mentioned above, the joint optimization of the quantizer and the input distribution of the channel is an interesting open problem. This applies to the asymptotic information-theoretic limit as well as to the design of finite blocklength quantizers. In both cases we have not yet found a viable method for solving the corresponding optimization problems.

- An extension of our scalar quantizer design algorithms to VQ would be worthwhile. Furthermore, it may be possible to perform the optimization of the quantizer based on samples of training data instead of the probability distributions.

- In the channel-optimized case there exists currently no approach for the numerical computation of the optimal rate-information trade-off. We believe that an extension of the iterative IB algorithm to the channel-optimized case is possible. This would allow us to compare the performance of our channel-optimized quantizers to the information-theoretic limit.

Quantization-Based Network Coding for the MARC

- The structure of the network code in the proposed scheme is very simple. An optimization of the network code can be expected to yield further performance improvements. Furthermore, a network code design which constrains the maximum degree of the vertices in the factor graph allows us to limit the decoding complexity at the relay.

- We assumed independence of the source data which is not always the case, e.g., in sensor networks. It should feasible to include source correlation in the factor graph of the overall network-channel code. The proposed iterative decoder then extends naturally to the correlated case.

- It is desirable to improve the proposed scheme such that the relay may adapt its coding strategy depending on the channel conditions. For example, in some cases it may be beneficial if the relay does not perform network coding. Finding a suitable set of coding strategies together with a practical adaptation policy is an open problem.

A

Proofs for Chapter 3

A.1 Proof of Lemma 3.2

To prove Lemma 3.2, we have to show that

$$L_{\mathsf{u}}^{c}(\boldsymbol{x}) = \log \frac{p(\boldsymbol{x}|\mathsf{u}=1)}{p(\boldsymbol{x}|\mathsf{u}=-1)} = \log \frac{p(L_{\mathsf{u}}(\boldsymbol{x})|\mathsf{u}=1)}{p(L_{\mathsf{u}}(\boldsymbol{x})|\mathsf{u}=-1)}. \tag{A.1}$$

To this end, we note that

$$\begin{aligned}
\int e^{kL_{\mathsf{u}}} p_{\mathsf{L_u}|\mathsf{u}}(L_{\mathsf{u}}|\mathsf{u}=1)\mathrm{d}L_{\mathsf{u}} &= \int e^{kL_{\mathsf{u}}(\boldsymbol{x})} p_{\mathbf{x}|\mathsf{u}}(\boldsymbol{x}|\mathsf{u}=1)\mathrm{d}\boldsymbol{x} && \text{(A.2)}\\
&= \int \left[\frac{\mathbb{P}\{\mathsf{u}=1|L_{\mathsf{u}}(\boldsymbol{x})\}}{\mathbb{P}\{\mathsf{u}=-1|L_{\mathsf{u}}(\boldsymbol{x})\}}\right]^{k} p_{\mathbf{x}|\mathsf{u}}(\boldsymbol{x}|\mathsf{u}=1)\mathrm{d}\boldsymbol{x} && \text{(A.3)}\\
&= e^{-L_{\mathsf{u}}^{a}} \int \left[\frac{\mathbb{P}\{\mathsf{u}=1|L_{\mathsf{u}}(\boldsymbol{x})\}}{\mathbb{P}\{\mathsf{u}=-1|L_{\mathsf{u}}(\boldsymbol{x})\}}\right]^{k+1} p_{\mathbf{x}|\mathsf{u}}(\boldsymbol{x}|\mathsf{u}=-1)\mathrm{d}\boldsymbol{x} && \text{(A.4)}\\
&= e^{-L_{\mathsf{u}}^{a}} \int e^{(k+1)L_{\mathsf{u}}(\boldsymbol{x})} p_{\mathbf{x}|\mathsf{u}}(\boldsymbol{x}|\mathsf{u}=-1)\mathrm{d}\boldsymbol{x} && \text{(A.5)}\\
&= e^{-L_{\mathsf{u}}^{a}} \int e^{(k+1)L_{\mathsf{u}}} p_{\mathsf{L_u}|\mathsf{u}}(L_{\mathsf{u}}|\mathsf{u}=-1)\mathrm{d}L_{\mathsf{u}}. && \text{(A.6)}
\end{aligned}$$

Since the equality of (A.2) and (A.6) holds for all $k \in \mathbb{Z}$ we have

$$e^{kL_{\mathsf{u}}} p_{\mathsf{L_u}|\mathsf{u}}(L_{\mathsf{u}}|\mathsf{u}=1) = e^{(k+1)L_{\mathsf{u}}-L_{\mathsf{u}}^{a}} p_{\mathsf{L_u}|\mathsf{u}}(L_{\mathsf{u}}|\mathsf{u}=-1). \tag{A.7}$$

Setting $k=0$ in (A.7), dividing by $p_{\mathsf{L_u}|\mathsf{u}}(L_{\mathsf{u}}|\mathsf{u}=-1)$, and taking the logarithm finally yields

$$\log \frac{p_{\mathsf{L_u}|\mathsf{u}}(L_{\mathsf{u}}|\mathsf{u}=1)}{p_{\mathsf{L_u}|\mathsf{u}}(L_{\mathsf{u}}|\mathsf{u}=-1)} = L_{\mathsf{u}} - L_{\mathsf{u}}^{a} = L_{\mathsf{u}}^{c}(\boldsymbol{x}). \tag{A.8}$$

A.2 Proof of Proposition 3.10

The mth derivative of $\big(1+\exp(L_{\mathsf{u}})\big)^{-2}$ can be shown to equal

$$\frac{\mathrm{d}^m}{\mathrm{d}L_{\mathsf{u}}^m}\left(\frac{1}{1+\exp(L_{\mathsf{u}})}\right)^2 = \sum_{k=1}^{m}(-1)^k d_{k,m}\frac{e^{kL_{\mathsf{u}}}}{(1+e^{L_{\mathsf{u}}})^{k+2}}, \tag{A.9}$$

with the coefficients $d_{k,m}$ as in (3.188). Using (A.9) we can write the Taylor series expansion of $\big(1+\exp(|L_{\mathsf{u}}|)\big)^{-2}$ around $L_{\mathsf{u}}=0$ as

$$\left(\frac{1}{1+\exp(|L_{\mathsf{u}}|)}\right)^2 = \frac{1}{4}+\sum_{m=1}^{\infty}\frac{|L_{\mathsf{u}}|^m}{m!}\sum_{k=1}^{m}\frac{(-1)^k}{2^{k+2}}d_{k,m}. \tag{A.10}$$

Taking the expectation of (A.10) with respect to the log-likelihood ratio yields (3.187).

A.3 Proof of (3.204)

We prove the inequality

$$\begin{aligned}\prod_{n=1}^{N}\big(1-(2-\alpha_n/K)P_{e,n}+(1-1/K)P_{e,n}^2\big)-\prod_{n=1}^{N}(1-P_{e,n})^2\\ \leq \frac{1}{K}\prod_{n=1}^{N}(1-(2-\alpha_n)P_{e,n})-\frac{1}{K}\prod_{n=1}^{N}(1-P_{e,n})^2\end{aligned} \tag{A.11}$$

by induction. The inequality in (A.11) holds for $N=1$. Furthermore, (A.11) is fulfilled with equality if $P_{e,n}=0$, $n=1,\ldots,N$, or $P_{e,n}=\alpha_n$, $n=1,\ldots,N$. In what follows, we therefore assume without loss of generality that $P_{e,N}\in(0,\alpha_N)$. We first rewrite (A.11) as

$$\prod_{n=1}^{N}\big(1-(2-\alpha_n/K)P_{e,n}+(1-1/K)P_{e,n}^2\big)\leq\frac{1}{K}\prod_{n=1}^{N}(1-(2-\alpha_n)P_{e,n})+\left(1-\frac{1}{K}\right)\prod_{n=1}^{N}(1-P_{e,n})^2. \tag{A.12}$$

The inequality in (A.12) is equivalent to

$$\begin{aligned}(1-(2-\alpha_N/K)P_{e,N}+(1-1/K)P_{e,N}^2)\prod_{n=1}^{N-1}\big(1-(2-\alpha_n/K)P_{e,n}+(1-1/K)P_{e,n}^2\big)\\ \leq\big(1-(2-\alpha_N/K)P_{e,N}+(1-1/K)P_{e,N}^2\big)\\ \times\left[\frac{1}{K}\prod_{n=1}^{N-1}(1-(2-\alpha_n)P_{e,n})+\left(1-\frac{1}{K}\right)\prod_{n=1}^{N-1}(1-P_{e,n})^2\right].\end{aligned} \tag{A.13}$$

Using (A.12) in (A.13) yields

$$\begin{aligned}(1-(2-\alpha_N/K)P_{e,N}\ &+(1-1/K)P_{e,N}^2\big)\\ &\times\left[\frac{1}{K}\prod_{n=1}^{N-1}(1-(2-\alpha_n)P_{e,n})+\left(1-\frac{1}{K}\right)\prod_{n=1}^{N-1}(1-P_{e,n})^2\right]\\ &\leq\frac{1}{K}\prod_{n=1}^{N}(1-(2-\alpha_n)P_{e,n})+\left(1-\frac{1}{K}\right)\prod_{n=1}^{N}(1-P_{e,n})^2. \qquad \text{(A.14)}\end{aligned}$$

To show that (A.11) holds we thus need to show that (A.14) holds. Sorting the terms in (A.14) yields

$$\left(1-\frac{1}{K}\right)\prod_{n=1}^{N-1}(1-P_{e,n})^2\leq\left(1-\frac{1}{K}\right)\prod_{n=1}^{N-1}(1-(2-\alpha_n)P_{e,n})\,. \qquad \text{(A.15)}$$

For $K=1$, (A.15) obviously holds with equality. For $K>1$, we have

$$\prod_{n=1}^{N-1}(1-P_{e,n})^2\leq\prod_{n=1}^{N-1}(1-(2-\alpha_n)P_{e,n})\,. \qquad \text{(A.16)}$$

The inequality in (A.16) holds if

$$(1-P_{e,n})^2\leq 1-(2-\alpha_n)P_{e,n},\quad n=1,\ldots,N-1. \qquad \text{(A.17)}$$

We note that (A.17) is equivalent to the true statement $P_{e,n}^2\leq\alpha_n P_{e,n}$ (cf. (3.202)). This concludes the proof of the weakened mean-square error (MSE) upper bound (3.204).

A.4 Proof of Proposition 3.11

In what follows, we prove that

$$\prod_{n=1}^{N}(1-P_{e,n})-\prod_{n=1}^{N}(1-P_{e,n})^2\geq\min_n\alpha_n^{-1}\left[\prod_{n=1}^{N}(1-(2-\alpha_n)P_{e,n})-\prod_{n=1}^{N}(1-P_{e,n})^2\right], \qquad \text{(A.18)}$$

where we have equality in (A.18) if and only if $P_{e,n}\equiv 0$. It is not hard to see that (A.18) is indeed fulfilled with equality if $P_{e,n}\equiv 0$. We therefore assume that at least one $P_{e,n}>0$ and we show that in this case

$$\prod_{n=1}^{N}(1-P_{e,n})-\prod_{n=1}^{N}(1-P_{e,n})^2>\min_n\alpha_n^{-1}\left[\prod_{n=1}^{N}(1-(2-\alpha_n)P_{e,n})-\prod_{n=1}^{N}(1-P_{e,n})^2\right]. \qquad \text{(A.19)}$$

Without loss of generality we assume that $P_{e,N} > 0$. We next rewrite (A.19) as follows:

$$\prod_{n=1}^{N}(1-P_{e,n}) + (\min_n \alpha_n^{-1} - 1)\prod_{n=1}^{N}(1-P_{e,n})^2 > \min_n \alpha_n^{-1} \prod_{n=1}^{N}(1-(2-\alpha_n)P_{e,n})\,. \tag{A.20}$$

We note that (A.20) holds for $N = 1$ if[1] $\alpha_N < 1$. Next, we rewrite (A.20) as

$$\begin{aligned}\min_n \alpha_n^{-1}\,(1-(2-\alpha_N)P_{e,N}) \prod_{n=1}^{N-1}(1-(2-\alpha_n)P_{e,n}) \\ < (1-(2-\alpha_N)P_{e,N}) \left[\prod_{n=1}^{N-1}(1-P_{e,n}) + (\min_n \alpha_n^{-1} - 1)\prod_{n=1}^{N-1}(1-P_{e,n})^2\right].\end{aligned} \tag{A.21}$$

To prove (A.19), we show that the following inequality holds:

$$\begin{aligned}(1-(2-\alpha_N)P_{e,N}) \left[\prod_{n=1}^{N-1}(1-P_{e,n}) + (\min_n \alpha_n^{-1} - 1)\prod_{n=1}^{N-1}(1-P_{e,n})^2\right] \\ < \prod_{n=1}^{N}(1-P_{e,n}) + (\min_n \alpha_n^{-1} - 1)\prod_{n=1}^{N}(1-P_{e,n})^2.\end{aligned} \tag{A.22}$$

By sorting the terms in (A.22), we obtain

$$P_{e,N}\,(1-\alpha_N)\prod_{n=1}^{N-1}(1-P_{e,n}) > P_{e,N}(\alpha_N - P_{e,N})(\min_n \alpha_n^{-1} - 1)\prod_{n=1}^{N-1}(1-P_{e,n})^2. \tag{A.23}$$

Cancelling terms in (A.23) yields

$$1-\alpha_N > (\alpha_N - P_{e,N})(\min_n \alpha_n^{-1} - 1)\prod_{n=1}^{N-1}(1-P_{e,n}). \tag{A.24}$$

We note that (A.24) holds if

$$\min_n \alpha_n^{-1} < \frac{1-P_{e,N}}{\alpha_N - P_{e,N}}. \tag{A.25}$$

The inequality in (A.25) is a true statement since $P_{e,N} > 0$. This concludes the proof of (3.220). Finally, (3.219) follows directly from this proof together with (3.204).

A.5 Proof of Theorem 3.12

We have

$$p(L_{\mathsf{u}};\mu) = \frac{1}{\sqrt{4\pi\mu}}\left[\exp\left(-\frac{1}{4\mu}(L_{\mathsf{u}}-\mu)^2\right)\mathbb{P}\{\mathsf{u}=1\} + \exp\left(-\frac{1}{4\mu}(L_{\mathsf{u}}+\mu)^2\right)\mathbb{P}\{\mathsf{u}=-1\}\right], \tag{A.26}$$

[1] We note that $P_{e,n} < 1$ can always be ensured if the detector simply guesses according to the prior probabilities of u_n. Therefore, we can assume without loss of generality that $\alpha_n < 1$, $n = 1, \ldots, N$.

where the error probability P_e is related to μ via the inverse Q-function as

$$\mu(P_e) = 2Q^{-2}(P_e). \tag{A.27}$$

The Fisher information in terms of μ equals

$$J(\mu) = -\int_{-\infty}^{\infty} \left(\frac{\partial^2}{\partial \mu^2} \log p(L_{\mathsf{u}}; \mu) \right) p(L_{\mathsf{u}}; \mu) \mathrm{d}L_{\mathsf{u}}. \tag{A.28}$$

To compute (A.28), we first compute $\frac{\partial^2}{\partial \mu^2} \log p(L_{\mathsf{u}}; \mu)$. For the first-order derivative we have

$$\frac{\partial}{\partial \mu} \log p(L_{\mathsf{u}}; \mu) = \frac{\sqrt{4\pi\mu}(2\sqrt{\mu} + \mu^{3/2} - L_{\mathsf{u}}^2/\sqrt{\mu})}{8\mu^2\sqrt{\pi}} = \frac{L_{\mathsf{u}}^2 - \mu(2+\mu)}{4\mu^2} \tag{A.29}$$

and thus the second-order derivative equals

$$\frac{\partial^2}{\partial \mu^2} \log p(L_{\mathsf{u}}; \mu) = -\frac{L_{\mathsf{u}}^2 - \mu}{2\mu^3}. \tag{A.30}$$

Using (A.30) in (A.28) and evaluating the integral yields

$$J(\mu) = \frac{1}{2\mu^3}\mu(2+\mu) - \frac{1}{2\mu^2} = \frac{1+\mu}{2\mu^2}. \tag{A.31}$$

Reparametrizing the Fisher information yields (cf. (2.39))

$$J(P_e) = J\big(\mu(P_e)\big) \left(\frac{\mathrm{d}\mu}{\mathrm{d}P_e} \right)^2. \tag{A.32}$$

The derivative $\mathrm{d}\mu/\mathrm{d}P_e$ equals

$$\frac{\mathrm{d}}{\mathrm{d}P_e} 2Q^{-2}(P_e) = -8\sqrt{\frac{\pi}{2}} Q^{-1}(P_e) \exp\big(Q^{-2}(P_e)/2\big). \tag{A.33}$$

In (A.33), we have used the relation [81, Section IV]

$$\frac{\mathrm{d}}{\mathrm{d}x} Q^{-1}(x) = -\sqrt{2\pi} \exp\big(Q^{-2}(x)/2\big). \tag{A.34}$$

Using (A.33), (A.31), and (A.27) in (A.32) yields

$$J(P_e) = \frac{1 + 2Q^{-2}(P_e)}{8Q^{-4}(P_e)} 32\pi Q^{-2}(P_e) \exp\big(Q^{-2}(P_e)\big) = \frac{4\pi \exp\big(Q^{-2}(P_e)\big)\big(1 + 2Q^{-2}(P_e)\big)}{Q^{-2}(P_e)}. \tag{A.35}$$

The Cramér-Rao lower bound is thus given as

$$\mathrm{MSE}_{\check{P}_e}(P_e) \geq \frac{Q^{-2}(P_e)}{4\pi \exp\big(Q^{-2}(P_e)\big)\big(1 + 2Q^{-2}(P_e)\big)}. \tag{A.36}$$

For K independent and identically distributed (iid) data samples the right-hand side of (A.36) is multiplied by $1/K$.

A.6 Proof of Theorem 3.13

An efficient estimator $\hat{P}_e^{\text{eff}}(L_{\mathsf{u}})$ for P_e exists if and only if (cf. (2.41))

$$\frac{\partial}{\partial P_e} \log p(L_{\mathsf{u}}; P_e) = J(P_e) \left(\hat{P}_e^{\text{eff}}(L_{\mathsf{u}}) - P_e \right). \tag{A.37}$$

Using (A.29) and (A.33), we can write the left-hand side of (A.37) as

$$\frac{\partial}{\partial P_e} \log p(L_{\mathsf{u}}; P_e) = \frac{\partial}{\partial \mu} \log p(L_{\mathsf{u}}; \mu) \frac{\mathrm{d}}{\mathrm{d}P_e} \mu(P_e) = \frac{1 + Q^{-2}(P_e) - L_{\mathsf{u}}^2}{Q^{-1}(P_e)} \sqrt{2\pi} \exp\bigl(Q^{-2}(P_e)/2\bigr). \tag{A.38}$$

Rewriting (A.37) in terms of $\hat{P}_e^{\text{eff}}(L_{\mathsf{u}})$ yields

$$\hat{P}_e^{\text{eff}}(L_{\mathsf{u}}) = P_e + \frac{1}{J(P_e)} \frac{\partial}{\partial P_e} \log p(L_{\mathsf{u}}; P_e). \tag{A.39}$$

Clearly, $\hat{P}_e^{\text{eff}}(L_{\mathsf{u}})$ is a valid estimator if it depends solely on the data L_{u}. Using (A.35) and (A.38), we further write (A.39) as

$$\hat{P}_e^{\text{eff}}(L_{\mathsf{u}}) = P_e + \frac{Q^{-1}(P_e) \left(1 + Q^{-2}(P_e) - L_{\mathsf{u}}^2\right)}{2\sqrt{2\pi} \exp\bigl(Q^{-2}(P_e)/2\bigr) \left(1 + 2Q^{-2}(P_e)\right)}. \tag{A.40}$$

However, (A.40) depends not only on L_{u} but also on the parameter P_e. Therefore, (A.40) is not a valid estimator. Since $\partial \log p(L_{\mathsf{u}}; P_e)/\partial P_e$ cannot be written as in (A.37), there exists no efficient estimator. This also holds if we consider the case of multiple iid samples.

B

Proofs for Chapter 4

B.1 Proof of Theorem 4.3

Due to (4.16) and (4.18), the optimal z equals

$$\mathsf{z} = a\mathsf{y} + \xi = ah\mathsf{x} + a\mathsf{w} + \xi, \tag{B.1}$$

where $\xi \sim \mathcal{N}(0,1)$ is independent of y and

$$a = \sqrt{\frac{[\beta(1+\gamma^{-1})^{-1} - 1]^+}{\sigma^2}}. \tag{B.2}$$

With $\mathsf{z} \sim \mathcal{N}(0, a^2(h^2P + \sigma^2) + 1)$ and $\mathsf{z}|y \sim \mathcal{N}(ay, 1)$, we can express the compression rate R in terms of β as

$$R(\beta) = I(\mathsf{y};\mathsf{z}) = h(\mathsf{z}) - h(\mathsf{z}|\mathsf{y}) = \frac{1}{2}\log_2\big(a^2(h^2P+\sigma^2)+1\big) = \frac{1}{2}\log_2^+\gamma(\beta-1), \tag{B.3}$$

where we have used (2.28) to compute the differential entropies. Using (B.3), we can express β in terms of R as follows:

$$\beta(R) = 1 + \frac{2^{2R}}{\gamma}. \tag{B.4}$$

From (B.1) it follows that $\mathsf{z}|x \sim \mathcal{N}(ahx, a^2\sigma^2 + 1)$, and thus the relevant information equals

$$I(\beta) = I(\mathsf{x};\mathsf{z}) = h(\mathsf{z}) - h(\mathsf{z}|\mathsf{x}) = \frac{1}{2}\log\frac{a^2(h^2P+\sigma^2)+1}{a^2\sigma^2+1} = R(\beta) - \frac{1}{2}\log\beta(1+\gamma^{-1})^{-1}. \tag{B.5}$$

Finally, using (B.4) in (B.5) yields the information-rate function $I(R)$ as in (4.24).

Let us next prove the properties of $I(R)$. To this end, we first rewrite $I(R)$ as follows:

$$I(R) = \frac{1}{2}\log_2 \frac{1+\gamma}{1+2^{-2R}\gamma}. \tag{B.6}$$

1. From the definition of strict concavity we have

$$I(\alpha R_1 + (1-\alpha)R_2) > \alpha I(R_1) + (1-\alpha)I(R_2), \quad 0 < \alpha < 1. \tag{B.7}$$

Without loss of generality we assume that $R_1 < R_2$ and define $R_\alpha \triangleq \alpha R_1 + (1-\alpha)R_2$. Then the following inequalities hold:

$$I(R_\alpha) > \alpha I(R_1) + (1-\alpha)I(R_2), \tag{B.8}$$

$$\log_2 \frac{1+\gamma}{1+2^{-2R_\alpha}\gamma} > \alpha \log_2 \frac{1+\gamma}{1+2^{-2R_1}\gamma} + (1-\alpha)\log_2 \frac{1+\gamma}{1+2^{-2R_2}\gamma}, \tag{B.9}$$

$$\log_2 \frac{1+2^{-2R_2}\gamma}{1+2^{-2R_\alpha}\gamma} > \log_2 \frac{1+2^{-2R_2}\gamma}{1+2^{-2R_1}\gamma} > \alpha \log_2 \frac{1+2^{-2R_2}\gamma}{1+2^{-2R_1}\gamma}, \tag{B.10}$$

$$2^{-2R_1} > 2^{-2R_\alpha}, \tag{B.11}$$

$$R_\alpha > R_1. \tag{B.12}$$

Due to our assumption $R_1 < R_2$, (B.12) is a true statement and, hence, $I(R)$ is strictly concave on $\mathbb{R}_+$.

2. The argument of the logarithm in (B.6) is strictly increasing in R and, since the logarithm is a strictly increasing function of its argument, $I(R)$ is strictly increasing in R.

3. From (4.24) it is obvious that $I(R) \leq R$ holds since the second term on the right-hand side of (4.24) is nonnegative. Similarly, $I(R) \leq C(\gamma)$ holds since the second term on the right-hand side of (4.25) is nonnegative.

4. Due to (4.24), we have $I(0) = 0$ and (4.25) immediately yields $\lim_{R\to\infty} I(R) = C(\gamma)$.

5. Taking the derivative of (4.24) with respect to R yields

$$\frac{\mathrm{d}I(R)}{\mathrm{d}R} = 1 - \frac{1}{2}\frac{1+\gamma}{2^{2R}+\gamma}\frac{2^{2R}2}{1+\gamma} = 1 - \frac{1}{1+2^{-2R}\gamma} = (1+2^{2R}\gamma^{-1})^{-1}. \tag{B.13}$$

B.2 Proof of Theorem 4.10

We first find the matrix $\boldsymbol{A}$ according to (4.18). To this end, we note that

$$\widetilde{\mathbf{y}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I} + \boldsymbol{\Gamma}) \qquad \text{and} \qquad \widetilde{\mathbf{y}}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{C}_{\mathbf{w}}^{-1/2}\boldsymbol{H}\boldsymbol{x}, \boldsymbol{I}). \tag{B.14}$$

Hence, we have

$$\boldsymbol{C}_{\widetilde{\mathbf{y}}|\boldsymbol{x}}\boldsymbol{C}_{\widetilde{\mathbf{y}}}^{-1} = (\boldsymbol{I}+\boldsymbol{\Gamma})^{-1} = \operatorname{diag}\{(1+\gamma_k)^{-1}\}_{k=1}^{n}. \tag{B.15}$$

Due to (B.15) we have $\boldsymbol{A} = \operatorname{diag}\{\alpha_k\}_{k=1}^{n}$, where (cf. (4.19))

$$\alpha_k = \sqrt{\left[\beta(1+\gamma_k^{-1})^{-1}-1\right]^+}. \tag{B.16}$$

Using (4.16), we have $\mathsf{z} = \boldsymbol{A}\widetilde{\mathbf{y}} + \boldsymbol{\xi}$, where

$$\boldsymbol{C}_{\mathsf{z}} = \boldsymbol{A}^2(\boldsymbol{I}+\boldsymbol{\Gamma}) + \boldsymbol{I} = \operatorname{diag}\left\{[\gamma_k(\beta-1)-1]^+ + 1\right\}_{k=1}^{n}. \tag{B.17}$$

Furthermore, we have the following conditional distributions:

$$\mathsf{z}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{C}_{\mathsf{w}}^{-1/2}\boldsymbol{H}\boldsymbol{x}, \boldsymbol{A}^2+\boldsymbol{I}) \quad \text{and} \quad \mathsf{z}|\widetilde{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{A}\widetilde{\boldsymbol{y}}, \boldsymbol{I}). \tag{B.18}$$

Using (B.17) and (B.18), we write the compression rate R in terms of β as

$$R(\beta) = I(\mathbf{y};\mathsf{z}) = I(\widetilde{\mathbf{y}};\mathsf{z}) = h(\mathsf{z}) - h(\mathsf{z}|\widetilde{\mathbf{y}}) = \frac{1}{2}\sum_{k=1}^{n}\log_2^+\gamma_k(\beta-1). \tag{B.19}$$

Similarly, for the relevant information we obtain

$$I(\beta) = I(\mathbf{x};\mathsf{z}) = h(\mathsf{z}) - h(\mathsf{z}|\mathbf{x}) = R(\beta) - \frac{1}{2}\sum_{k=1}^{n}\log_2^+\beta(1+\gamma_k^{-1})^{-1}. \tag{B.20}$$

To eliminate β from (B.20) we first observe that only the first

$$\ell(\beta) = \sum_{k=1}^{n}\mathbb{1}\{\beta > \beta_{c,k}\} \tag{B.21}$$

terms contribute to the sums in (B.19) and (B.20), where the critical values of β are given as

$$\beta_{c,k} = 1+\gamma_k^{-1}. \tag{B.22}$$

Without loss of generality, we assume that the mode signal-to-noise ratios (SNRs) are sorted in descending order, i.e., we have $\gamma_1 \geq \cdots \geq \gamma_n$. We note that $R(\beta) = 0$ and $I(\beta) = 0$ if $\beta \leq 1+\gamma_1^{-1}$. Assuming that $\beta > 1+\gamma_1^{-1}$, we can implicitly express β using (B.21) in (B.19) as follows:

$$\beta = 1 + \frac{2^{2R(\beta)/\ell(\beta)}}{\overline{\gamma}_{\ell(\beta)}}. \tag{B.23}$$

Here, we use $\overline{a}_n$ to denote the geometric mean of the n numbers $a_1,\ldots,a_n$, i.e., we have

$$\overline{\gamma}_{\ell(\beta)} \triangleq \prod_{i=1}^{\ell(\beta)}\gamma_i^{1/\ell(\beta)}. \tag{B.24}$$

Using (B.23) in (B.20) yields

$$I(\beta) = \begin{cases} R(\beta) - \dfrac{1}{2}\displaystyle\sum_{k=1}^{\ell(\beta)} \log_2 \left[\left(1 + \dfrac{2^{2R(\beta)/\ell(\beta)}}{\overline{\gamma}_{\ell(\beta)}} \right) (1+\gamma_k^{-1})^{-1} \right], & \beta > 1+\gamma_1^{-1} \\ 0, & \beta \leq 1+\gamma_1^{-1} \end{cases}. \tag{B.25}$$

We can equivalently express (B.21) in terms of R as in (4.39) by replacing (B.22) with the corresponding critical rates

$$R_{c,k} = \frac{1}{2}\sum_{i=1}^{k} \log_2 \frac{\gamma_i}{\gamma_k}. \tag{B.26}$$

This allows us to write the information-rate function $I(R)$ as

$$I(R) = \begin{cases} R - \dfrac{1}{2}\displaystyle\sum_{k=1}^{\ell(R)} \log_2 \frac{2^{2R/\ell(R)}\gamma_k/\overline{\gamma}_{\ell(R)} + \gamma_k}{1+\gamma_k}, & R > 0 \\ 0, & R = 0 \end{cases} \tag{B.27}$$

$$= R - \frac{1}{2}\sum_{k=1}^{n} \log_2 \frac{2^{2R_k(R)} + \gamma_k}{1+\gamma_k}, \tag{B.28}$$

with the rate allocation $R_k(R)$ as in (4.38).

The properties 3 to 5 of $I(R)$ can be verified from (4.36)-(4.39) (see also Appendix B.1). The properties 1 and 2 follow from the proof of Lemma 4.19.

B.3 Proof of Theorem 4.14

We first use (B.19), (B.20), and (B.21) to write the relevant information as follows:

$$I(\beta) = \frac{1}{2}\sum_{k=1}^{\ell(\beta)} \log_2 \frac{\beta - 1}{\beta}(1+\gamma_k). \tag{B.29}$$

Next, assuming that $\beta > 1+\gamma_1^{-1}$ (with the mode SNRs sorted in descending order), we use (B.29) to implicitly express β as

$$\beta = \left(1 - \frac{2^{2I(\beta)/\ell(\beta)}}{\overline{(1+\gamma)}_{\ell(\beta)}} \right)^{-1}. \tag{B.30}$$

Using (B.30) in (B.20) yields

$$R(\beta) = \begin{cases} I(\beta) + \dfrac{1}{2}\displaystyle\sum_{k=1}^{\ell(\beta)} \log_2 \frac{(1+\gamma_k^{-1})^{-1}}{1 - \frac{2^{2I(\beta)/\ell(\beta)}}{\overline{(1+\gamma)}_{\ell(\beta)}}}, & \beta > 1+\gamma_1^{-1} \\ 0, & \beta \leq 1+\gamma_1^{-1} \end{cases}. \tag{B.31}$$

We next express (B.21) in terms of I as in (4.45) by replacing (B.22) with the corresponding critical relevant information

$$I_{c,k} = \frac{1}{2}\sum_{i=1}^{k} \log_2 \frac{1+\gamma_i}{1+\gamma_k}. \tag{B.32}$$

This allows us to write the rate-information function as

$$R(I) = \begin{cases} \frac{1}{2}\sum_{k=1}^{\ell(I)} \log_2 \frac{\gamma_k}{2^{-2I/\ell(I)}\overline{(1+\gamma)}_{\ell(I)} - 1}, & I > 0 \\ 0, & I = 0 \end{cases} \tag{B.33}$$

$$= \frac{1}{2}\sum_{k=1}^{n} \log_2 \frac{\gamma_k}{2^{-2I_k(I)}(1+\gamma_k) - 1}, \tag{B.34}$$

with the information allocation $I_k(I)$ as in (4.44).

The properties 3 to 5 of $R(I)$ can be verified from (4.36)-(4.39). The properties 1 and 2 follow from the proof of Lemma 4.19 and the fact that $R(I)$ is the inverse of $I(R)$ (cf. Corollary 4.15).

B.4 Proof of Lemma 4.17

We have $\check{\mathbf{y}} = \boldsymbol{F}\widetilde{\mathbf{y}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_{\check{\mathbf{y}}})$ with $\boldsymbol{F} = \operatorname{diag}\{f_k\}_{k=1}^n$, where $\boldsymbol{C}_{\check{\mathbf{y}}} = \boldsymbol{F}(\boldsymbol{I}+\boldsymbol{\Gamma})\boldsymbol{F}$. mean-square-error-optimal compression of $\check{\mathbf{y}}$ is modeled using a "forward channel" (cf. [8, Sec. 4.3]). Hence, we can write the quantized version of $\check{\mathbf{y}}$ as

$$\mathbf{z} = \boldsymbol{B}\check{\mathbf{y}} + \boldsymbol{\eta}, \tag{B.35}$$

where $\boldsymbol{B} = \operatorname{diag}\{b_k\}_{k=1}^n$ and $\boldsymbol{\eta} \sim \mathcal{N}\big(\mathbf{0}, \operatorname{diag}\{b_k D_k\}_{k=1}^n\big)$ is independent of $\check{\mathbf{y}}$. Here, we have

$$b_k = 1 - \frac{D_k}{\omega_k}, \tag{B.36}$$

with $\omega_k = f_k^2(1+\gamma_k)$, $k = 1, \ldots, n$, and the reverse waterfilling rate allocation [20, Sec. 10.3]

$$R(\theta, \boldsymbol{F}) = \sum_{k=1}^{n} R_k(\theta, \boldsymbol{F}), \qquad R_k(\theta, \boldsymbol{F}) = \frac{1}{2}\log_2^+ \frac{\omega_k}{\theta}, \tag{B.37}$$

$$D(\theta, \boldsymbol{F}) = \sum_{k=1}^{n} D_k(\theta, \boldsymbol{F}), \qquad D_k(\theta, \boldsymbol{F}) = \min\{\theta, \omega_k\}, \tag{B.38}$$

where $\theta > 0$ is the waterlevel. Without loss of generality, we assume that the ω_k's are sorted in descending order, i.e., we have $\omega_1 \geq \cdots \geq \omega_n$. Due to (B.35) and (B.36), we have (recall that $\widetilde{\mathbf{y}} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{C}_{\mathbf{w}}^{-1/2}(\boldsymbol{H}\mathbf{x} + \mathbf{w})$)

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_{\mathbf{z}}) \quad \text{and} \quad \mathbf{z}|\boldsymbol{x} \sim \mathcal{N}\big(\boldsymbol{B}\boldsymbol{F}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{C}_{\mathbf{w}}^{-1/2}\boldsymbol{H}\boldsymbol{x}, \boldsymbol{C}_{\mathbf{z}|\boldsymbol{x}}\big), \tag{B.39}$$

where

$$\boldsymbol{C}_{\mathbf{z}} = \operatorname{diag}\{b_k^2 f_k^2(1+\gamma_k) + b_k D_k\}_{k=1}^n, \tag{B.40}$$

$$\boldsymbol{C}_{\mathbf{z}|\boldsymbol{x}} = \operatorname{diag}\{b_k^2 f_k^2 + b_k D_k\}_{k=1}^n. \tag{B.41}$$

Next, we express the relevant information using (B.36), (B.40), and (B.41) as follows:

$$I^{\mathrm{RD}}(\theta, \boldsymbol{F}) = I(\mathbf{x};\mathbf{z}) = h(\mathbf{z}) - h(\mathbf{z}|\mathbf{x}) = \frac{1}{2}\sum_{k=1}^{n} \log_2 \frac{1+\gamma_k}{1+D_k(\theta,\boldsymbol{F})\gamma_k/\omega_k}. \tag{B.42}$$

Using (B.38), we can rewrite (B.42) as

$$I^{\mathrm{RD}}(\theta, \boldsymbol{F}) = \frac{1}{2}\sum_{k=1}^{l(\theta,\boldsymbol{F})} \log_2 \frac{1+\gamma_k}{1+\theta\gamma_k/\omega_k}, \tag{B.43}$$

since only the first

$$l(\theta, \boldsymbol{F}) = \sum_{k=1}^{n} \mathbb{1}\{\theta < \omega_k\} \tag{B.44}$$

terms contribute to the sum in (B.42). We note that $I^{\mathrm{RD}}(\theta, \boldsymbol{F}) = 0$ for $\theta \geq \omega_1$. Assuming that $\theta < \omega_1$, we can implicitly express the waterlevel θ using (B.44) and (B.37) as follows:

$$\theta = 2^{-2R(\theta,\boldsymbol{F})/l(\theta,\boldsymbol{F})}\overline{\omega}_{l(\theta,\boldsymbol{F})}. \tag{B.45}$$

With (B.45) we obtain

$$I^{\mathrm{RD}}(\theta, \boldsymbol{F}) = \begin{cases} \dfrac{1}{2}\displaystyle\sum_{k=1}^{l(\theta,\boldsymbol{F})} \log_2 \frac{1+\gamma_k}{1+2^{-2R(\theta,\boldsymbol{F})/l(\theta,\boldsymbol{F})}\gamma_k\overline{\omega}_{l(\theta,\boldsymbol{F})}/\omega_k}, & \theta < \omega_1 \\ 0, & \theta \geq \omega_1 \end{cases}. \tag{B.46}$$

Expressing (B.44) in terms of R by replacing ω_k with the corresponding critical rates $R_{c,k}^{\mathrm{RD}}$ (cf. (4.55)) yields

$$l(R, \boldsymbol{F}) = \sum_{k=1}^{n} \mathbb{1}\{R > R_{c,k}^{\mathrm{RD}}\}. \tag{B.47}$$

With (B.37), (B.45), and (B.47) we obtain the rate allocation

$$R_k^{\mathrm{RD}}(R, \boldsymbol{F}) = \begin{cases} \left[\dfrac{R}{l(R,\boldsymbol{F})} + \dfrac{1}{2}\log_2 \dfrac{\omega_k}{\overline{\omega}_{l(R,\boldsymbol{F})}}\right]^+, & R > 0 \\ 0, & R = 0 \end{cases}. \tag{B.48}$$

Using (B.48) in (B.46) yields (4.52).

B.5 Proof of Lemma 4.19

To prove that $I^{\mathrm{RD}}(R, \boldsymbol{F})$ is concave in R for arbitrary mode SNRs if and only if (4.68) holds, we show that $I^{\mathrm{RD}}(R, \boldsymbol{F})$ is nondecreasing and has a nonincreasing derivative. From the waterfilling representation (4.64) and (4.65), it is not hard to see that $I^{\mathrm{RD}}(R, \boldsymbol{F})$ is nondecreasing in R (since R is nonincreasing in θ). The derivative of $I^{\mathrm{RD}}(R, \boldsymbol{F})$ equals

$$\frac{\mathrm{d}I^{\mathrm{RD}}(R, \boldsymbol{F})}{\mathrm{d}R} = \frac{\frac{\mathrm{d}I^{\mathrm{RD}}(\theta, \boldsymbol{F})}{\mathrm{d}\theta}}{\frac{\mathrm{d}R(\theta, \boldsymbol{F})}{\mathrm{d}\theta}} \tag{B.49}$$

$$= \frac{1}{l(\theta, \boldsymbol{F})} \sum_{k=1}^{l(\theta, \boldsymbol{F})} \frac{\theta\gamma_k/\omega_k}{1 + \theta\gamma_k/\omega_k} \tag{B.50}$$

$$= I'^{\mathrm{RD}}(\theta, \boldsymbol{F}), \tag{B.51}$$

where θ is chosen such that $\frac{1}{2}\sum_{k=1}^{n} \log_2^+ \frac{\omega_k}{\theta} = R$. The derivative of $I^{\mathrm{RD}}(R, \boldsymbol{F})$ is nonincreasing in R if (B.51) is nondecreasing in θ. In the following let $\theta_1 \leq \theta_2$. If $l(\theta_1, \boldsymbol{F}) = l(\theta_2, \boldsymbol{F})$, we have $I'^{\mathrm{RD}}(\theta_1, \boldsymbol{F}) \leq I'^{\mathrm{RD}}(\theta_2, \boldsymbol{F})$, since the term in the sum in (B.50) is increasing in θ. If $l(\theta_1, \boldsymbol{F}) > l(\theta_2, \boldsymbol{F})$, we have $I'^{\mathrm{RD}}(\theta_1, \boldsymbol{F}) \leq I'^{\mathrm{RD}}(\theta_2, \boldsymbol{F})$ if and only if the quantities γ_k/ω_k, $k = 1, \ldots, n$, are sorted in descending order. This is because (B.50) is an arithmetic mean of $l(\theta, \boldsymbol{F})$ terms which is decreasing if and only if additional smaller terms are added. Therefore, the filter coefficients have to satisfy

$$\frac{f_{k+1}^2}{f_k^2} \geq \frac{\frac{\gamma_{k+1}}{1+\gamma_{k+1}}}{\frac{\gamma_k}{1+\gamma_k}}, \qquad k = 1, \ldots, n-1. \tag{B.52}$$

From (B.52) it is evident that the sign of the filter coefficients is irrelevant. For the ω_k's, (B.52) entails

$$\frac{\omega_{k+1}}{\omega_k} \geq \frac{\gamma_{k+1}}{\gamma_k}, \qquad k = 1, \ldots, n-1. \tag{B.53}$$

The optimal filter satisfies (B.52) and (B.53) with equality since $\boldsymbol{F} = \boldsymbol{F}^\star$ entails $\omega_k = \gamma_k$. Particularizing (B.52) to $\boldsymbol{F} = \boldsymbol{W}^\rho$ for $\rho \geq 0$ yields

$$\xi_k^{2\rho-1} \geq 1, \qquad k = 1, \ldots, n-1, \qquad \text{with} \qquad \xi_k = \frac{\frac{\gamma_{k+1}}{1+\gamma_{k+1}}}{\frac{\gamma_k}{1+\gamma_k}}, \tag{B.54}$$

where the mode SNRs γ_k are sorted in descending order (to ensure that the ω_k's are sorted in descending order). Hence, we have $\xi_k \leq 1$ and thus $\xi_k^{2\rho-1} \geq 1$ if and only if $\rho \leq 1/2$. This concludes the proof of the first part of Lemma 4.19.

For the derivative of $I^{\mathrm{RD}}(R, \boldsymbol{F}^\star)$, we have (cf. (B.50))

$$I'^{\mathrm{RD}}(\theta, \boldsymbol{F}^\star) = \frac{\theta}{1+\theta}. \tag{B.55}$$

We note that the derivative in (B.55) is continuous, nondecreasing, and concave in θ and thus

$\frac{\mathrm{d}I^{\mathrm{RD}}(R,\boldsymbol{F}^\star)}{\mathrm{d}R}$ is continuous, nonincreasing, and convex in R (since θ is inversely proportional to R). Furthermore, $I^{\mathrm{RD}}(R,\boldsymbol{F}^\star)$ does not depend on the mode SNRs. Next, let

$$I_+'^{\mathrm{RD}}(\omega_l,\boldsymbol{F}) = \lim_{\theta\downarrow\omega_l} I'^{\mathrm{RD}}(\theta,\boldsymbol{F}) = \frac{1}{l-1}\sum_{k=1}^{l-1}\frac{\gamma_k\frac{\omega_l}{\omega_k}}{1+\gamma_k\frac{\omega_l}{\omega_k}}, \tag{B.56}$$

$$I_-'^{\mathrm{RD}}(\omega_l,\boldsymbol{F}) = \lim_{\theta\uparrow\omega_l} I'^{\mathrm{RD}}(\theta,\boldsymbol{F}) = \frac{1}{l}\sum_{k=1}^{l}\frac{\gamma_k\frac{\omega_l}{\omega_k}}{1+\gamma_k\frac{\omega_l}{\omega_k}}. \tag{B.57}$$

We note that $I_+'^{\mathrm{RD}}(\omega_l,\boldsymbol{F}) = I_-'^{\mathrm{RD}}(\omega_l,\boldsymbol{F})$, $l=2,\ldots,n$, if and only if $\boldsymbol{F}=\boldsymbol{F}^\star$ (recall that in this case $\omega_k=\gamma_k$). If $\boldsymbol{F}\neq\boldsymbol{F}^\star$, we have

$$\frac{I_-'^{\mathrm{RD}}(\omega_l,\boldsymbol{F})}{I_+'^{\mathrm{RD}}(\omega_l,\boldsymbol{F})} = \frac{1}{l}\left(l-1+\frac{1}{I_+'^{\mathrm{RD}}(\omega_l,\boldsymbol{F})}\frac{\gamma_l}{1+\gamma_l}\right)\neq 1, \qquad l=2,\ldots,n, \tag{B.58}$$

i.e., $\frac{\mathrm{d}I^{\mathrm{RD}}(R,\boldsymbol{F}^\star)}{\mathrm{d}R}$ is discontinuous at the critical rates. This concludes the proof of the second part of Lemma 4.19.

B.6 Proof of Lemma 4.20

The critical rates are given as

$$R_{c,k}^{\mathrm{RD}}(\boldsymbol{I}) = \frac{1}{2}\sum_{i=1}^{k}\log_2\frac{1+\gamma_i}{1+\gamma_k}, \tag{B.59}$$

$$R_{c,k} = \frac{1}{2}\sum_{i=1}^{k}\log_2\frac{\gamma_i}{\gamma_k}, \tag{B.60}$$

$$R_{c,k}^{\mathrm{RD}}(\boldsymbol{W}) = \frac{1}{2}\sum_{i=1}^{k}\log_2\frac{\gamma_i^2}{1+\gamma_i}\frac{1+\gamma_k}{\gamma_k^2}. \tag{B.61}$$

Therefore, we have

$$\frac{R_{c,k}^{\mathrm{RD}}(\boldsymbol{I})+R_{c,k}^{\mathrm{RD}}(\boldsymbol{W})}{2} = \frac{1}{4}\sum_{i=1}^{k}\log_2\frac{1+\gamma_i}{1+\gamma_k}\frac{\gamma_i^2}{1+\gamma_i}\frac{1+\gamma_k}{\gamma_k^2} = \frac{1}{2}\sum_{i=1}^{k}\log_2\frac{\gamma_i}{\gamma_k} = R_{c,k}. \tag{B.62}$$

The inequality $R_{c,k}^{\mathrm{RD}}(\boldsymbol{I})\leq R_{c,k}$ holds since each term in (B.59) is smaller than (or equal to) the corresponding term in (B.60). Specifically,

$$\frac{1+\gamma_i}{1+\gamma_k}\leq\frac{\gamma_i}{\gamma_k}, \quad i=1,\ldots,k, \tag{B.63}$$

yields $\gamma_i \geq \gamma_k$ which is a true statement since the mode SNRs are sorted in descending order and $i = 1, \ldots, k$. Similarly, $R_{c,k} \leq R_{c,k}^{\mathrm{RD}}(\boldsymbol{W})$ holds since

$$\frac{\gamma_i}{\gamma_k} \leq \frac{\gamma_i^2}{1+\gamma_i} \frac{1+\gamma_k}{\gamma_k^2}, \quad i = 1, \ldots, k, \tag{B.64}$$

again yields the true statement $\gamma_i \geq \gamma_k$. Together with (4.39) and (4.54), the inequalities $R_{c,k}^{\mathrm{RD}}(\boldsymbol{I}) \leq R_{c,k} \leq R_{c,k}^{\mathrm{RD}}(\boldsymbol{W})$ for the critical rates yield the inequalities $l(R, \boldsymbol{I}) \geq \ell(R) \geq l(R, \boldsymbol{W})$ for the number of active modes.

C

Proofs for Chapter 5

C.1 Proof of Proposition 5.3

To prove Proposition 5.3, we first rewrite $I(\boldsymbol{g})$ and $I\big(\boldsymbol{g}, \boldsymbol{h}(\boldsymbol{x})\big)$ as follows:

$$I(\boldsymbol{g}) = \sum_{\boldsymbol{x}\in\mathcal{X}^l} \sum_{z\in\mathcal{Z}} p(\boldsymbol{x}|z)p(z) \log \frac{p(\boldsymbol{x}|z)}{p(\boldsymbol{x})}, \tag{C.1}$$

$$I\big(\boldsymbol{g}, \boldsymbol{h}(\boldsymbol{x})\big) = \sum_{\boldsymbol{x}\in\mathcal{X}^l} \sum_{z\in\mathcal{Z}} p(\boldsymbol{x}|z)p(z) \log \frac{h_z(\boldsymbol{x})}{p(\boldsymbol{x})}, \tag{C.2}$$

where

$$p(\boldsymbol{x}|z) = \frac{\displaystyle\int_{g_{z-1}}^{g_z} p(\boldsymbol{x}|y)p(y)\mathrm{d}y}{\displaystyle\int_{g_{z-1}}^{g_z} p(y)\mathrm{d}y} \qquad \text{and} \qquad p(z) = \int_{g_{z-1}}^{g_z} p(y)\mathrm{d}y. \tag{C.3}$$

Next, we compute the difference between (C.1) and (C.2). We have

$$I(\boldsymbol{g}) - I\big(\boldsymbol{g}, \boldsymbol{h}(\boldsymbol{x})\big) = \sum_{\boldsymbol{x}\in\mathcal{X}^l} \sum_{z\in\mathcal{Z}} p(\boldsymbol{x}|z)p(z) \log \frac{p(\boldsymbol{x}|z)}{h_z(\boldsymbol{x})} \tag{C.4}$$

$$= D\big(p(\boldsymbol{x}|z)p(z) \| h_z(\boldsymbol{x})p(z)\big) \tag{C.5}$$

$$\geq 0. \tag{C.6}$$

We note that (C.6) is due to the information inequality (2.22). Thus, we have shown that $I(\boldsymbol{g}) \geq I\big(\boldsymbol{g}, \boldsymbol{h}(\boldsymbol{x})\big)$ and due to (C.5) we have $I(\boldsymbol{g}) = I\big(\boldsymbol{g}, \boldsymbol{h}(\boldsymbol{x})\big)$ if and only if $h_z(\boldsymbol{x}) = p(\boldsymbol{x}|z)$. This concludes the proof since $p(\boldsymbol{x}|z)$ in (C.3) is equal to $h_i^\star(\boldsymbol{x})$ in (5.21).

C.2 Proof of Proposition 5.4

To prove the convergence of our algorithm to a locally optimal solution of (5.20), we show that the updates (5.32) and (5.34) do not decrease the value of the objective function. Specifically, we establish the following inequalities:

$$I\big(\boldsymbol{g}^{(i)}, \boldsymbol{h}^{(i)}(x)\big) \leq I\big(\boldsymbol{g}^{(i+1)}, \boldsymbol{h}^{(i)}(x)\big) \leq I\big(\boldsymbol{g}^{(i+1)}, \boldsymbol{h}^{(i+1)}(x)\big). \tag{C.7}$$

Since the objective function is upper bounded as $I\big(\boldsymbol{g}^{(i)}, \boldsymbol{h}^{(i)}(x)\big) \leq H(\mathsf{x})$, the inequalities in (C.7) imply convergence to a local optimum as $i \to \infty$.

To prove that the first inequality in (C.7) holds, we show that $\boldsymbol{g}^{(i+1)}$ is unique and corresponds to a local maximum if the log-likelihood ratio (LLR) $L_{\mathsf{x}}(y)$ is strictly increasing in y. To this end, we note that (cf. (5.33))

$$\frac{\partial^2}{\partial g_j \partial g_m} I\big(\boldsymbol{g}, \boldsymbol{h}(x)\big) = 0, \qquad j \neq m, \tag{C.8}$$

and therefore we have to show that

$$\frac{\partial^2}{\partial g_j^2} I\big(\boldsymbol{g}^{(i+1)}, \boldsymbol{h}^{(i)}(x)\big) < 0, \qquad j = 1, \ldots, n-1. \tag{C.9}$$

The inequality in (C.9) is equivalent to the following inequality (we suppress the iteration index in what follows):

$$\log \frac{1+e^{-L_{j+1}}}{1+e^{-L_j}} \frac{\mathrm{d}}{\mathrm{d}g_j} \mathbb{P}\{\mathsf{x}{=}1|\mathsf{y}{=}g_j\} + \log \frac{1+e^{L_{j+1}}}{1+e^{L_j}} \frac{\mathrm{d}}{\mathrm{d}g_j} \mathbb{P}\{\mathsf{x}{=}{-1}|\mathsf{y}{=}g_j\} < 0. \tag{C.10}$$

Next, we express the derivative of the posterior probability $\mathbb{P}\{\mathsf{x}{=}x|\mathsf{y}{=}g_j\}$ as

$$\frac{\mathrm{d}}{\mathrm{d}g_j} p_{\mathsf{x}|\mathsf{y}}(x|g_j) = \frac{\mathrm{d}}{\mathrm{d}g_j} \frac{1}{1+e^{-xL_{\mathsf{x}}(g_j)}} = \frac{xL'(g_j)e^{-xL_{\mathsf{x}}(g_j)}}{\big(1+e^{-xL_{\mathsf{x}}(g_j)}\big)^2}. \tag{C.11}$$

Using (C.11) in (C.10) yields

$$\log \frac{1+e^{-L_{j+1}}}{1+e^{-L_j}} < \log \frac{1+e^{L_{j+1}}}{1+e^{L_j}} \tag{C.12}$$

which is in turn equivalent to

$$L_j < L_{j+1}. \tag{C.13}$$

Hence, the stationary point $\boldsymbol{g}^{(i+1)}$ is a local maximum if the LLRs L_j, $j = 1, \ldots, n$, (cf. (5.32)) are sorted in ascending order. We next show that (C.13) holds if the LLR $L_{\mathsf{x}}(y)$ is

strictly increasing in y. Writing (C.13) more explicitly yields

$$\frac{\displaystyle\int_{g_{j-1}^{+}}^{g_j^{-}} \frac{1}{1+e^{-L_{\mathsf{x}}(y)}} p(y)\mathrm{d}y}{\displaystyle\int_{g_{j-1}^{+}}^{g_j^{-}} \frac{1}{1+e^{L_{\mathsf{x}}(y)}} p(y)\mathrm{d}y} < \frac{\displaystyle\int_{g_j^{+}}^{g_{j+1}^{-}} \frac{1}{1+e^{-L_{\mathsf{x}}(y)}} p(y)\mathrm{d}y}{\displaystyle\int_{g_j^{+}}^{g_{j+1}^{-}} \frac{1}{1+e^{L_{\mathsf{x}}(y)}} p(y)\mathrm{d}y}. \tag{C.14}$$

Here, the notation $\int_{a^+}^{b^-}$ implies integration over the interval $[a, b)$. Next, we lower bound the right-hand side of (C.14) as follows:

$$\frac{\displaystyle\int_{g_j^{+}}^{g_{j+1}^{-}} \frac{1}{1+e^{-L_{\mathsf{x}}(y)}} p(y)\mathrm{d}y}{\displaystyle\int_{g_j^{+}}^{g_{j+1}^{-}} \frac{1}{1+e^{L_{\mathsf{x}}(y)}} p(y)\mathrm{d}y} \geq \frac{\dfrac{1}{1+e^{-L_{\mathsf{x}}(g_j^{+})}} \displaystyle\int_{g_j^{+}}^{g_{j+1}^{-}} p(y)\mathrm{d}y}{\dfrac{1}{1+e^{L_{\mathsf{x}}(g_j^{+})}} \displaystyle\int_{g_j^{+}}^{g_{j+1}^{-}} p(y)\mathrm{d}y} = e^{L_{\mathsf{x}}(g_j^{+})}. \tag{C.15}$$

Similarly, the left-hand side of (C.14) can be upper bounded as

$$\frac{\displaystyle\int_{g_{j-1}^{+}}^{g_j^{-}} \frac{1}{1+e^{-L_{\mathsf{x}}(y)}} p(y)\mathrm{d}y}{\displaystyle\int_{g_{j-1}^{+}}^{g_j^{-}} \frac{1}{1+e^{L_{\mathsf{x}}(y)}} p(y)\mathrm{d}y} \leq \frac{\dfrac{1}{1+e^{-L_{\mathsf{x}}(g_j^{-})}} \displaystyle\int_{g_{j-1}^{+}}^{g_j^{-}} p(y)\mathrm{d}y}{\dfrac{1}{1+e^{L_{\mathsf{x}}(g_j^{-})}} \displaystyle\int_{g_{j-1}^{+}}^{g_j^{-}} p(y)\mathrm{d}y} = e^{L_{\mathsf{x}}(g_j^{-})}. \tag{C.16}$$

Using (C.15) and (C.16) in (C.14) yields

$$e^{L_{\mathsf{x}}(g_j^{-})} < e^{L_{\mathsf{x}}(g_j^{+})} \tag{C.17}$$

which is a true statement if $L_{\mathsf{x}}(y)$ is strictly increasing in y. In this case, $\boldsymbol{g}^{(i+1)}$ is indeed a local maximum of the objective function. To conclude that $I\big(\boldsymbol{g}^{(i)}, \boldsymbol{h}^{(i)}(x)\big) \leq I\big(\boldsymbol{g}^{(i+1)}, \boldsymbol{h}^{(i)}(x)\big)$, we require that $\boldsymbol{g}^{(i+1)}$ is the only stationary point, i.e., the only solution of (5.34). We note that $L_{\mathsf{x}}(y)$ is injective since it is strictly increasing and hence there is at most one stationary point. To see that there must exist at least one solution of (5.34), note that the right-hand side of (5.34) can be bounded as follows:

$$L_j \leq \log \frac{\log \dfrac{1+e^{L_{j+1}}}{1+e^{L_j}}}{\log \dfrac{1+e^{-L_j}}{1+e^{-L_{j+1}}}} \leq L_{j+1}, \qquad j = 1, \ldots, n-1. \tag{C.18}$$

Furthermore, we can bound the LLRs L_1 and L_n using (5.32) as $L_{\mathsf{x}}(g_0) \leq L_1$ and $L_{\mathsf{x}}(g_n) \geq L_n$, respectively. Hence, for each $j \in \{1, \ldots, n-1\}$ there must exist at least one g_j such that $L_{\mathsf{x}}(g_j)$ equals the right-hand side of (5.34). We have thus established that $I\big(\boldsymbol{g}^{(i)}, \boldsymbol{h}^{(i)}(x)\big) \leq I\big(\boldsymbol{g}^{(i+1)}, \boldsymbol{h}^{(i)}(x)\big)$. The inequality $I\big(\boldsymbol{g}^{(i+1)}, \boldsymbol{h}^{(i)}(x)\big) \leq I\big(\boldsymbol{g}^{(i+1)}, \boldsymbol{h}^{(i+1)}(x)\big)$ follows immediately from Proposition 5.3. This concludes the proof of Proposition 5.4.

D

Moments of the Normal Distribution

In this appendix, we derive expressions for the (raw) moments, the central moments, the (raw) absolute moments, and the central absolute moments of a normal (Gaussian) random variable $\mathsf{x} \sim \mathcal{N}(\mu, \sigma^2)$ with mean $\mu = \mathbb{E}\{\mathsf{x}\}$ and variance $\sigma^2 = \mathbb{E}\{\mathsf{x}^2\} - \mu^2$. In Section D.1, we introduce several special functions which we use to express the moments of x. In Section D.2, we present the resulting formulas for the moments of x and the corresponding derivations are given in Section D.3.

D.1 Preliminaries

In the following we list the definitions of subsequently used special functions (cf., e.g., [70]):

- *Gamma function:*

$$\Gamma(z) \triangleq \int_0^\infty t^{z-1} e^{-t} \mathrm{d}t. \tag{D.1}$$

- *Rising factorial:*

$$z^{\overline{n}} \triangleq \frac{\Gamma(z+n)}{\Gamma(z)} \tag{D.2}$$

$$= z(z+1)\cdots(z+n-1), \quad n \in \mathbb{N}_0. \tag{D.3}$$

- *Double factorial:*

$$z!! \triangleq \sqrt{\frac{2^{z+1}}{\pi}}\,\Gamma\Big(\frac{z}{2}+1\Big) \tag{D.4}$$

$$= z \cdot (z-2) \cdot \ldots \cdot 3 \cdot 1, \quad z \in \mathbb{N} \text{ odd}. \tag{D.5}$$

- *Kummer's confluent hypergeometric functions:*

$$\Phi(\alpha, \gamma; z) \triangleq \sum_{n=0}^{\infty} \frac{\alpha^{\overline{n}}}{\gamma^{\overline{n}}} \frac{z^n}{n!}. \tag{D.6}$$

- *Tricomi's confluent hypergeometric functions:*

$$\Psi(\alpha, \gamma; z) \triangleq \frac{\Gamma(1-\gamma)}{\Gamma(\alpha-\gamma+1)} \Phi(\alpha, \gamma; z) + \frac{\Gamma(\gamma-1)}{\Gamma(\alpha)} z^{1-\gamma} \Phi(\alpha-\gamma+1, 2-\gamma; z). \tag{D.7}$$

- *Parabolic cylinder functions:*

$$D_\nu(z) \triangleq 2^{\nu/2} e^{-z^2/4} \left[\frac{\sqrt{\pi}}{\Gamma\big(\frac{1-\nu}{2}\big)} \Phi\Big(-\frac{\nu}{2}, \frac{1}{2}; \frac{z^2}{2}\Big) - \frac{\sqrt{2\pi} z}{\Gamma\big(-\frac{\nu}{2}\big)} \Phi\Big(\frac{1-\nu}{2}, \frac{3}{2}; \frac{z^2}{2}\Big) \right]. \tag{D.8}$$

D.2 Results

In this section we summarize formulas for the raw/central (absolute) moments of a normal random variable $\mathsf{x} \sim \mathcal{N}(\mu, \sigma^2)$. The formulas for $\mathbb{E}\{\mathsf{x}^\nu\}$, $\mathbb{E}\{(\mathsf{x}-\mu)^\nu\}$, $\mathbb{E}\{|\mathsf{x}|^\nu\}$, and $\mathbb{E}\{|\mathsf{x}-\mu|^\nu\}$ hold for $\nu > -1$ unless stated otherwise. Note that $j = \sqrt{-1}$ denotes the imaginary unit.

Raw Moments.

$$\mathbb{E}\{\mathsf{x}^\nu\} = (j\sigma)^\nu \exp\Big(-\frac{\mu^2}{4\sigma^2}\Big)\, D_\nu\Big(-j\frac{\mu}{\sigma}\Big) \tag{D.9}$$

$$= (j\sigma)^\nu 2^{\nu/2} \left[\frac{\sqrt{\pi}}{\Gamma\left(\frac{1-\nu}{2}\right)} \Phi\left(-\frac{\nu}{2}, \frac{1}{2}; -\frac{\mu^2}{2\sigma^2}\right) + j\frac{\mu}{\sigma} \frac{\sqrt{2\pi}}{\Gamma\left(-\frac{\nu}{2}\right)} \Phi\left(\frac{1-\nu}{2}, \frac{3}{2}; -\frac{\mu^2}{2\sigma^2}\right) \right] \tag{D.10}$$

$$= (j\sigma)^\nu 2^{\nu/2} \cdot \begin{cases} \Psi\left(-\frac{\nu}{2}, \frac{1}{2}; -\frac{\mu^2}{2\sigma^2}\right), & \mu \leq 0 \\ \Psi^*\left(-\frac{\nu}{2}, \frac{1}{2}; -\frac{\mu^2}{2\sigma^2}\right), & \mu > 0 \end{cases} \tag{D.11}$$

$$= \begin{cases} \sigma^\nu 2^{\nu/2} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi}} \Phi\left(-\frac{\nu}{2}, \frac{1}{2}; -\frac{\mu^2}{2\sigma^2}\right), & \nu \in \mathbb{N}_0 \text{ even} \\ \mu\sigma^{\nu-1} 2^{(\nu+1)/2} \frac{\Gamma\left(\frac{\nu}{2}+1\right)}{\sqrt{\pi}} \Phi\left(\frac{1-\nu}{2}, \frac{3}{2}; -\frac{\mu^2}{2\sigma^2}\right), & \nu \in \mathbb{N}_0 \text{ odd} \end{cases}. \tag{D.12}$$

Central Moments.

$$\mathbb{E}\{(\mathsf{x}-\mu)^\nu\} = (j\sigma)^\nu 2^{\nu/2} \frac{\sqrt{\pi}}{\Gamma\left(\frac{1-\nu}{2}\right)} \tag{D.13}$$

$$= (j\sigma)^\nu 2^{\nu/2} \cos(\pi\nu/2) \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi}} \tag{D.14}$$

$$= \left(1 + (-1)^\nu\right) \sigma^\nu 2^{\nu/2-1} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi}} \tag{D.15}$$

$$= \begin{cases} \sigma^\nu (\nu-1)!!, & \nu \in \mathbb{N}_0 \text{ even} \\ 0, & \nu \in \mathbb{N}_0 \text{ odd} \end{cases}. \tag{D.16}$$

Raw Absolute Moments.

$$\mathbb{E}\{|\mathsf{x}|^\nu\} = \sigma^\nu 2^{\nu/2} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi}} \Phi\left(-\frac{\nu}{2}, \frac{1}{2}; -\frac{\mu^2}{2\sigma^2}\right). \tag{D.17}$$

Central Absolute Moments.

$$\mathbb{E}\{|\mathsf{x}-\mu|^\nu\} = \sigma^\nu 2^{\nu/2} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi}}. \tag{D.18}$$

D.3 Derivations

In this section we give derivations for the results presented above. We use the following two identities (which hold for $\gamma \in \mathbb{R}$ and $\nu > -1$) to express the moments in terms of special functions (cf. [48, Sec. 3.462]):

$$\int_{-\infty}^{\infty} (-jx)^\nu e^{-x^2 + j\gamma x} \mathrm{d}x = \sqrt{2^{-\nu}\pi} e^{-\gamma^2/8} D_\nu\left(\frac{\gamma}{\sqrt{2}}\right), \tag{D.19}$$

$$\int_0^\infty x^\nu e^{-x^2-\gamma x} \mathrm{d}x = 2^{-(\nu+1)/2} \Gamma(\nu+1) e^{\gamma^2/8} D_{-\nu-1}\left(\frac{\gamma}{\sqrt{2}}\right). \tag{D.20}$$

Raw Moments. Using (D.19), we obtain (D.9) from the definition of $\mathbb{E}\{\mathsf{x}^\nu\}$ as follows:

$$\mathbb{E}\{\mathsf{x}^\nu\} = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} x^\nu \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \mathrm{d}x \tag{D.21}$$

$$= \sqrt{\frac{2^\nu \sigma^{2\nu}}{\pi}} \exp\Big(-\frac{\mu^2}{2\sigma^2}\Big) \int_{-\infty}^{\infty} x^\nu \exp\Big(-x^2 + x\frac{\mu}{\sigma}\sqrt{2}\Big)\,\mathrm{d}x \tag{D.22}$$

$$\overset{(D.19)}{=} (j\sigma)^\nu \exp\Big(-\frac{\mu^2}{4\sigma^2}\Big)\, D_\nu\Big(-j\frac{\mu}{\sigma}\Big)\,. \tag{D.23}$$

Central Moments. Equation (D.13) follows from (D.9) with $\Phi(\alpha,\gamma;0)=1$ and, hence,

$$D_\nu(0) = 2^{\nu/2}\frac{\sqrt{\pi}}{\Gamma\big(\frac{1-\nu}{2}\big)}. \tag{D.24}$$

To obtain (D.14) from (D.13) we use the identity [48, Sec. 8.334]

$$\Gamma\Big(\frac{1+\nu}{2}\Big)\,\Gamma\Big(\frac{1-\nu}{2}\Big) = \frac{\pi}{\cos(\pi\nu/2)}. \tag{D.25}$$

Then (D.15) follows from (D.14) by noting that

$$\cos(\pi\nu/2) = \frac{1+\exp(j\pi\nu)}{2\exp(j\pi\nu/2)} = \frac{1+(-1)^\nu}{2j^\nu}. \tag{D.26}$$

Raw Absolute Moments. Using (D.20), we obtain (D.17) from the definition of $\mathbb{E}\big\{|\mathsf{x}|^\nu\big\}$ as follows:

$$\mathbb{E}\big\{|\mathsf{x}|^\nu\big\} = \frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{\infty}|x|^\nu \exp\Big(-\frac{1}{2\sigma^2}(x-\mu)^2\Big)\,\mathrm{d}x \tag{D.27}$$

$$= \sqrt{\frac{2^\nu\sigma^{2\nu}}{\pi}}\,\exp\Big(-\frac{\mu^2}{2\sigma^2}\Big)\,\Big[\int_0^\infty x^\nu\exp\Big(-x^2 - x\frac{\mu}{\sigma}\sqrt{2}\Big)\,\mathrm{d}x + \int_0^\infty x^\nu\exp\Big(-x^2+x\frac{\mu}{\sigma}\sqrt{2}\Big)\,\mathrm{d}x\Big] \tag{D.28}$$

$$\overset{(D.20)}{=} \sqrt{\frac{2^\nu\sigma^{2\nu}}{\pi}}\,\exp\Big(-\frac{\mu^2}{4\sigma^2}\Big)\,2^{-(\nu+1)/2}\Gamma(\nu+1)\big(D_{-\nu-1}(\mu/\sigma)+D_{-\nu-1}(-\mu/\sigma)\big) \tag{D.29}$$

$$= \sqrt{\frac{\sigma^{2\nu}}{2^\nu}}\,\exp\Big(-\frac{\mu^2}{2\sigma^2}\Big)\,\frac{\Gamma(\nu+1)}{\Gamma(\nu/2+1)}\,\Phi\Big(\frac{\nu+1}{2},\frac{1}{2};\frac{\mu^2}{2\sigma^2}\Big) \tag{D.30}$$

$$= \sqrt{\frac{2^\nu\sigma^{2\nu}}{\pi}}\,\exp\Big(-\frac{\mu^2}{2\sigma^2}\Big)\,\Gamma\Big(\frac{\nu+1}{2}\Big)\,\Phi\Big(\frac{\nu+1}{2},\frac{1}{2};\frac{\mu^2}{2\sigma^2}\Big) \tag{D.31}$$

$$= \sigma^\nu 2^{\nu/2}\frac{\Gamma\big(\frac{\nu+1}{2}\big)}{\sqrt{\pi}}\Phi\Big(-\frac{\nu}{2},\frac{1}{2};-\frac{\mu^2}{2\sigma^2}\Big)\,, \tag{D.32}$$

In (D.31), we have used Kummer's transformation $\Phi(\alpha,\gamma;z) = e^z\Phi(\gamma-\alpha,\gamma;-z)$ [48, Sec. 9.212] to obtain (D.32).

Central Absolute Moments. Equation (D.18) follows from (D.17) with $\Phi(\alpha,\gamma;0)=1$.

List of Abbreviations

3GPP	**3**rd **G**eneration **P**artnership **P**roject
AF	**a**mplify-and-**f**orward
APP	**a p**osteriori **p**robability
AWGN	**a**dditive **w**hite **G**aussian **n**oise
BCJR	**B**ahl, **C**ocke, **J**elinek, and **R**aviv
BEC	**b**inary **e**rasure **c**hannel
BER	**b**it **e**rror **r**ate
BICM	**b**it-**i**nterleaved **c**oded **m**odulation
BP	**b**elief **p**ropagation
BPSK	**b**inary **p**hase-**s**hift **k**eying
BSC	**b**inary **s**ymmetric **c**hannel
COVQ	**c**hannel-**o**ptimized **v**ector **q**uantization
CRLB	**C**ramér-**R**ao **l**ower **b**ound
CSI	**c**hannel **s**tate **i**nformation
DF	**d**ecode-and-**f**orward
DMC	**d**iscrete **m**emoryless **c**hannel
DVB	**d**igital **v**ideo **b**roadcasting
FER	**f**rame **e**rror **r**ate
GIB	**G**aussian **i**nformation **b**ottleneck
HMM	**h**idden **M**arkov **m**odel
IB	**i**nformation **b**ottleneck
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
iid	**i**ndependent and **i**dentically **d**istributed
IoT	**i**nternet **o**f **t**hings
LBG	**L**inde, **B**uzo, and **G**ray
LDPC	**l**ow-**d**ensity **p**arity-**c**heck
LLR	**l**og-**l**ikelihood **r**atio
LTE-A	**L**ong **T**erm **E**volution-**A**dvanced
MAP	**m**aximum **a p**osteriori

MARC	**m**ultiple-**a**ccess **r**elay **c**hannel
ML	**m**aximum **l**ikelihood
MOE	**m**aximum **o**utput **e**ntropy
MSE	**m**ean-**s**quare **e**rror
MVU	**m**inimum-**v**ariance **u**nbiased
pdf	**p**robability **d**ensity **f**unction
pmf	**p**robability **m**ass **f**unction
QAM	**q**uadrature **a**mplitude **m**odulation
RD	**r**ate-**d**istortion
SAC	**s**tuck-**a**t **c**hannel
SNR	**s**ignal-to-**n**oise **r**atio
VQ	**v**ector **q**uantization

Bibliography

[1] 3GPP, *TR 36.814 Further advancements for E-UTRA: Physical layer aspects.* www.3gpp.org, TR 36.814 v9.0.0, March 2010.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, *Network information flow*, IEEE Trans. Inf. Theory, 46 (2000), pp. 1204–1216.

[3] S. Arimoto, *An algorithm for computing the capacity of arbitrary discrete memoryless channels*, IEEE Trans. Inf. Theory, 18 (1972), pp. 14–20.

[4] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, *Optimal decoding of linear codes for minimizing symbol error rate*, IEEE Trans. Inf. Theory, 20 (1974), pp. 284–287.

[5] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains*, Ann. Math. Stat., 41 (1970), pp. 164–171.

[6] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, *Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding*, IEEE Trans. Inf. Theory, 44 (1998), pp. 909–926.

[7] C. Benkeser, A. Burg, T. Cupaiuolo, and Q. Huang, *Design and optimization of an HSDPA turbo decoder ASIC*, IEEE J. Solid-State Circuits, 44 (2009), pp. 98–106.

[8] T. Berger, *Rate Distortion Theory*, Prentice Hall, Englewood Cliffs (NJ), 1971.

[9] C. Berrou, A. Glavieux, and P. Thitimajshime, *Near Shannon limit error-correcting coding and decoding: Turbo-codes*, in Proc. IEEE Int. Conf. Commun. (ICC), May 1993, pp. 1064–1070.

[10] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont (MA), 2nd ed., 1999.

[11] R. E. Blahut, *Computation of channel capacity and rate-distortion functions*, IEEE Trans. Inf. Theory, 18 (1972), pp. 460–473.

[12] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge Univ. Press, Cambridge (UK), 2004.

[13] R. P. Brent, *Algorithms for Minimization without Derivatives*, Prentice Hall, Englewood Cliffs (NJ), 1973.

[14] D. Burshtein, V. D. Pietra, D. Kanevsky, and A. Nadas, *Minimum impurity partitions*, Ann. Stat., 20 (1992), pp. 1637–1646.

[15] N. Champaneria, T. Moon, and J. Gunther, *A soft-output stack algorithm*, in Proc. 40th Asilomar Conf. Signals, Systems, Computers, Oct. 2006, pp. 2195–2199.

[16] L. Chebli, C. Hausl, G. Zeitler, and R. Koetter, *Cooperative uplink of two mobile stations with network coding based on the WiMax LDPC code*, in Proc. IEEE Global Commun. Conf. (GLOBECOM), Nov. 2009.

[17] G. Chechik, A. Globerson, N. Tishby, and Y. Weiss, *Information bottleneck for Gaussian variables*, J. Mach. Learn. Res., 6 (2005), pp. 165–188.

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

[19] T. M. Cover and A. E. Gamal, *Capacity theorems for the relay channel*, IEEE Trans. Inf. Theory, 25 (1979), pp. 572–584.

[20] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, New York (NY), 2nd ed., Sept. 2006.

[21] H. Cramér, *Mathematical methods of statistics*, Princeton Univ. Press, Princeton (NJ), 1946.

[22] M. Danieli, S. Forchhammer, J. Andersen, L. Christensen, and S. Christensen, *Maximum mutual information vector quantization of log-likelihood ratios for memory efficient HARQ implementations*, in Proc. Data Compression Conf. (DCC), March 2010, pp. 30–39.

[23] ETSI, *Digital video broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2).* EN 302 307, V1.2.1, Aug. 2009. Available online at http://www.etsi.org/deliver/etsi_en/302300_302399/302307/01.02.01_60/en_302307v010201p.pdf.

[24] R. Fano, *A heuristic discussion of probabilistic decoding*, IEEE Trans. Inf. Theory, 9 (1963), pp. 64–74.

[25] N. Farvardin, *A study of vector quantization for noisy channels*, IEEE Trans. Inf. Theory, 36 (1990), pp. 799–809.

[26] N. Farvardin and V. Vaishampayan, *On the performance and complexity of channel-optimized vector quantizers*, IEEE Trans. Inf. Theory, 37 (1991), pp. 155–160.

[27] P. Fertl, J. Jaldén, and G. Matz, *Performance assessment of MIMO-BICM demodulators based on mutual information*, IEEE Trans. Signal Process., 60 (2012), pp. 1366–1382.

[28] V. Franz and J. Anderson, *Concatenated decoding with a reduced-search BCJR algorithm*, IEEE J. Sel. Areas Commun., 16 (1998), pp. 186–195.

[29] R. G. Gallager, *Low-density parity-check codes*, IRE Trans. Inf. Theory, 8 (1962), pp. 21–28.

[30] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Boston: Kluwer, 1992.

[31] R. Gilad-Bachrach, A. Navot, and N. Tishby, *An information theoretic tradeoff between complexity and accuracy*, in Learning Theory and Kernel Machines, Springer, 2003, pp. 595–609.

[32] A. Globerson and N. Tishby, *On the optimality of the Gaussian information bottleneck curve*, tech. rep., The Hebrew Univ. of Jerusalem, Feb. 2004.

[33] N. Goertz, *On the iterative approximation of optimal joint source-channel decoding*, IEEE J. Sel. Areas Commun., 19 (2001), pp. 1662–1670.

[34] A. Goldsmith, *Wireless Communications*, Cambridge Univ. Press, Cambridge (UK), 2005.

[35] R. M. Gray, *Source Coding Theory*, Kluwer Academic Publishers, Boston (MA), 1990.

[36] J. Hagenauer, *Soft is better than hard*, in Communications and Cryptography, R. E. Blahut, D. J. Costello, U. Maurer, and T. Mittelholzer, eds., Kluwer Academic Publishers, 1994, pp. 155–171.

[37] J. Hagenauer, *A soft-in/soft-out list sequential (LISS) decoder for turbo schemes*, in Proc. IEEE Int. Symp. Inf. Theory (ISIT), June 2003, pp. 382–382.

[38] J. Hagenauer, *The EXIT chart — Introduction to extrinsic information transfer in iterative processing*, in Proc. Eur. Signal Process. Conf. (EUSIPCO), Sept. 2004, pp. 1541–1548.

[39] J. Hagenauer and P. Hoeher, *A Viterbi algorithm with soft-decision outputs and its applications*, in Proc. IEEE Global Commun. Conf. (GLOBECOM), vol. 3, Nov. 1989, pp. 1680–1686.

[40] J. Hagenauer, E. Offer, and L. Papke, *Iterative decoding of binary block and convolutional codes*, IEEE Trans. Inf. Theory, 42 (1996), pp. 429–445.

[41] A. Hatefi, R. Visoz, and A. Berthet, *Joint channel-network turbo coding for the non-orthogonal multiple access relay channel*, in Proc. 21st Int. Symp. Personal Indoor and Mobile Radio Commun. (PIMRC), Sept. 2010, pp. 408–413.

[42] C. Hausl and P. Dupraz, *Joint network-channel coding for the multiple-access relay channel*, Proc. 3rd Annu. Conf. Sensor and Ad Hoc Commun. and Networks (SECON), 3 (2006), pp. 817–822.

[43] P. Hoeher, I. Land, and U. Sorger, *Log-likelihood values and Monte Carlo simulation – some fundamental results*, in Proc. 2nd Int. Symp. Turbo Codes and Related Topics, Sept. 2000, pp. 43–46.

[44] IEEE LAN/MAN Standards Committee, *IEEE 802.11-2012: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, 2012. Available online at http://standards.ieee.org/getieee802/download/802.11-2012.pdf.

[45] IEEE LAN/MAN Standards Committee, *IEEE 802.16-2012: Air interface for broadband wireless access systems*, 2012. Available online at http://standards.ieee.org/getieee802/download/802.16-2012.pdf.

[46] O. Iscan and C. Hausl, *Iterative network and channel decoding for the relay channel with multiple sources*, in Proc. IEEE Veh. Technol. Conf. (VTC), Sept. 2011.

[47] J. Jaldén, P. Fertl, and G. Matz, *On the generalized mutual information of BICM systems with approximate demodulation*, in Proc. IEEE Inf. Theory Workshop (ITW), Jan. 2010.

[48] A. Jeffrey and D. Zwillinger, eds., *Table of integrals, series, and products*, Academic Press, 6th ed., 2000.

[49] F. Jelinek, *Fast sequential decoding algorithm using a stack*, IBM J. Res. Dev., 13 (1969), pp. 675–685.

[50] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, Englewood Cliffs (NJ), 1993.

[51] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*, Prentice Hall, Upper Saddle River (NJ), 1998.

[52] G. Kramer, M. Gastpar, and P. Gupta, *Cooperative strategies and capacity theorems for relay networks*, IEEE Trans. Inf. Theory, 51 (2005), pp. 3037–3063.

[53] M. Krein and D. Milman, *On extreme points of regular convex sets*, Stud. Math., 9 (1940), pp. 133–138.

[54] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, *Factor graphs and the sum-product algorithm*, IEEE Trans. Inf. Theory, 47 (2001), pp. 498–519.

[55] B. Kurkoski and H. Yagi, *Concatenation of a discrete memoryless channel and a quantizer*, in Proc. IEEE Inf. Theory Workshop (ITW), Jan. 2010.

[56] B. Kurkoski and H. Yagi, *Finding the capacity of a quantized binary-input DMC*, in Proc. IEEE Int. Symp. Inf. Theory (ISIT), July 2012, pp. 686–690.

[57] I. Land and P. Hoeher, *New results on Monte Carlo bit error simulation based on the a posteriori log-likelihood ratio*, in Proc. 3rd Int. Symp. Turbo Codes and Related Topics, Sept. 2003, pp. 531–534.

[58] I. Land, P. Hoeher, and S. Gligorević, *Computation of symbol-wise mutual information in transmission systems with LogAPP decoders and application to EXIT charts*, in Proc. 5th Int. ITG Conf. Source and Channel Coding (SCC), Jan. 2004, pp. 195–202.

[59] I. Land and J. Huber, *Information combining*, Found. Trends Commun. Inf. Theory, 3 (2006), pp. 227–330.

[60] I. Land, S. Huettinger, P. Hoeher, and J. Huber, *Bounds on information combining*, IEEE Trans. Inf. Theory, 51 (2005), pp. 612–619.

[61] G. Lechner and J. Sayir, *Improved sum-min decoding for irregular LDPC codes*, in Proc. 4th Int. Symp. Turbo Codes and Related Topics, April 2006.

[62] X. Li and J. Ritcey, *Bit-interleaved coded modulation with iterative decoding*, IEEE Commun. Lett., 1 (1997), pp. 169–171.

[63] R. Lidl and H. Niederreiter, *Finite Fields*, Cambridge Univ. Press, Cambridge (UK), 2nd ed., 2008.

[64] S. Lin and D. J. Costello, *Error Control Coding*, Prentice Hall, Upper Saddle River (NJ), 2nd ed., 2004.

[65] Y. Linde, A. Buzo, and R. M. Gray, *An algorithm for vector quantizer design*, IEEE Trans. Commun., 28 (1980), pp. 84–95.

[66] S. Lloyd, *Least squares quantization in PCM*, IEEE Trans. Inf. Theory, 28 (1982), pp. 129–137.

[67] H.-A. Loeliger, *A posteriori probabilities and performance evaluation of trellis codes*, in Proc. IEEE Int. Symp. Inf. Theory (ISIT), June 1994, p. 335.

[68] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. Kschischang, *The factor graph approach to model-based signal processing*, Proc. IEEE, 95 (2007), pp. 1295–1322.

[69] D. J. C. MacKay, *Good error correcting codes based on very sparse matrices*, IEEE Trans. Inf. Theory, 45 (1999), pp. 399–431.

[70] W. Magnus, F. Oberhettinger, and R. P. Soni, *Formulas and Theorems for the Special Functions of Mathematical Physics*, Springer, 3rd ed., 1966.

[71] J. Max, *Quantization for minimum distortion*, IEEE Trans. Inf. Theory, 6 (1960), pp. 7–12.

[72] M. Meidlinger, A. Winkelbauer, and G. Matz, *On the relation between the Gaussian information bottleneck and MSE-optimal rate-distortion quantization*, in Proc. IEEE Workshop Statistical Signal Process. (SSP), June 2014, pp. 89–92.

[73] D. Messerschmitt, *Quantizing for maximum output entropy*, IEEE Trans. Inf. Theory, 17 (1971), pp. 612–612.

[74] V. Muralidharan and B. Rajan, *Physical layer network coding for the K-user multiple access relay channel*, IEEE Trans. Wireless Commun., 12 (2013), pp. 3107–3119.

[75] B. Nazer and M. Gastpar, *Computing over multiple-access channels with connections to wireless network coding*, in Proc. IEEE Int. Symp. Inf. Theory (ISIT), July 2006, pp. 1354–1358.

[76] B. Nazer and M. Gastpar, *Compute-and-forward: Harnessing interference through structured codes*, IEEE Trans. Inf. Theory, 57 (2011), pp. 6463–6486.

[77] B. Nazer and M. Gastpar, *Reliable physical layer network coding*, Proc. IEEE, 99 (2011), pp. 438–460.

[78] C. Novak, P. Fertl, and G. Matz, *Quantization for soft-output demodulators in bit-interleaved coded modulation systems*, in Proc. IEEE Int. Symp. Inf. Theory (ISIT), July 2009, pp. 1070–1074.

[79] C. Novak and G. Matz, *Low-complexity MIMO-BICM receivers with imperfect channel state information: Capacity-based performance comparison*, in Proc. 11th Int. Workshop on Signal Process. Advances in Wireless Commun. (SPAWC), June 2010.

[80] C. Novak, C. Studer, A. Burg, and G. Matz, *The effect of unreliable LLR storage on the performance of MIMO-BICM*, in Proc. 44th Asilomar Conf. Signals, Systems, Computers, Nov. 2010, pp. 736–740.

[81] J. R. Philip, *The function inverfc θ*, Austr. J. Phys., 13 (1960), pp. 13–20.

[82] H. V. Poor, *An Introduction to Signal Detection and Estimation*, Springer, New York, 1988.

[83] P. Popovski and H. Yomo, *The anti-packets can increase the achievable throughput of a wireless multi-hop network*, in Proc. IEEE Int. Conf. Commun. (ICC), vol. 9, June 2006, pp. 3885–3890.

[84] C. R. Rao, *Information and accuracy attainable in the estimation of statistical parameters*, Bulletin of the Calcutta Mathematical Society, 37 (1945), pp. 81–91.

[85] W. Rave, *Quantization of log-likelihood ratios to maximize mutual information*, IEEE Signal Process. Lett., 16 (2009), pp. 283–286.

[86] T. J. Richardson and R. L. Urbanke, *The capacity of low-density parity-check codes under message-passing decoding*, IEEE Trans. Inf. Theory, 47 (2001), pp. 599–618.

[87] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*, Cambridge Univ. Press, Cambridge (UK), 2008.

[88] S. Rosati, S. Tomasin, M. Butussi, and B. Rimoldi, *LLR compression for BICM systems using large constellations*, IEEE Trans. Commun., 61 (2013), pp. 2864–2875.

[89] C. Roth, C. Benkeser, C. Studer, G. Karakonstantis, and A. Burg, *Data mapping for unreliable memories*, in Proc. 50th Annu. Allerton Conf. Commun., Control, Comput., Oct. 2012, pp. 679–685.

[90] W. Ryan and S. Lin, *Channel Codes: Classical and Modern*, Cambridge Univ. Press, Cambridge (UK), 2009.

[91] S. Schwandter, P. Fertl, C. Novak, and G. Matz, *Log-likelihood ratio clipping in MIMO-BICM systems: Information geometric analysis and impact on system capacity*, in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), April 2009, pp. 2433–2436.

[92] A. Sendonaris, E. Erkip, and B. Aazhang, *User cooperation diversity — Part I: System description*, IEEE Trans. Commun., 51 (2003), pp. 1927–1938.

[93] A. Sendonaris, E. Erkip, and B. Aazhang, *User cooperation diversity — Part II: Implementation aspects and performance analysis*, IEEE Trans. Commun., 51 (2003), pp. 1939–1948.

[94] C. E. Shannon, *A mathematical theory of communication*, Bell Syst. Tech. J., 27 (1948), pp. 379–423, 623–656.

[95] S. ten Brink, *Convergence behavior of iteratively decoded parallel concatenated codes*, IEEE Trans. Commun., 49 (2001), pp. 1727–1737.

[96] R. Thobaben and I. Land, *Blind quality estimation for corrupted source signals based on a-posteriori probabilities*, in Proc. IEEE Int. Symp. Inf. Theory (ISIT), June 2004, p. 302.

[97] N. Tishby, F. Pereira, and W. Bialek, *The information bottleneck method*, in Proc. 37th Annu. Allerton Conf. Commun., Control, Comput., Sept. 1999, pp. 368–377.

[98] M. Tüchler, R. Koetter, and A. C. Singer, *Turbo equalization: Principles and new results*, IEEE Trans. Commun., 50 (2002), pp. 754–767.

[99] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I: Detection, Estimation, and Linear Modulation Theory*, Wiley, New York (NY), 1968.

[100] A. J. Viterbi, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, IEEE Trans. Inf. Theory, 13 (1967), pp. 260–269.

[101] T. Wang and G. Giannakis, *Complex field network coding for multiuser cooperative communications*, IEEE J. Sel. Areas Commun., 26 (2008), pp. 561–571.

[102] N. Wiberg, *Codes and Decoding on General Graphs*, PhD thesis, Dept. of Electrical Engineering, Linköping, Sweden, 1996. Linköping studies in Science and Technology. Dissertation No. 440.

[103] A. Winkelbauer, *Moments and absolute moments of the normal distribution*, Sept. 2012. Available online at http://arxiv.org/abs/1209.4340.

[104] A. Winkelbauer, S. Farthofer, and G. Matz, *The rate-information trade-off for Gaussian vector channels*, in Proc. IEEE Int. Symp. Inf. Theory (ISIT), July 2014, pp. 2849–2853.

[105] A. Winkelbauer, N. Goertz, and G. Matz, *Compress-and-forward in the multiple-access relay channel: with or without network coding?*, in Proc. 7th Int. Symp. Turbo Codes and Related Topics, Aug. 2012, pp. 131–135.

[106] A. Winkelbauer and G. Matz, *On efficient soft-input soft-output encoding of convolutional codes*, in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), May 2011.

[107] A. Winkelbauer and G. Matz, *Blind estimation of bit and block error probabilities using soft information*, in Proc. 50th Annu. Allerton Conf. Commun., Control, Comput., Oct. 2012, pp. 1278–1285.

[108] A. Winkelbauer and G. Matz, *Joint network-channel coding for the asymmetric multiple-access relay channel*, in Proc. IEEE Int. Conf. Commun. (ICC), June 2012, pp. 2485–2489.

[109] A. Winkelbauer and G. Matz, *Joint network-channel coding in the multiple-access relay channel: Beyond two sources*, in Proc. 5th Int. Symp. Commun., Control, Signal Process. (ISCCSP), May 2012.

[110] A. Winkelbauer and G. Matz, *Rate-information-optimal Gaussian channel output compression*, in Proc. 48th Annu. Conf. Inf. Sci. Syst. (CISS), March 2014.

[111] A. Winkelbauer, G. Matz, and A. Burg, *Channel-optimized vector quantization with mutual information as fidelity criterion*, in Proc. 47th Asilomar Conf. Signals, Systems, Computers, Nov. 2013, pp. 851–855.

[112] H. Witsenhausen and A. Wyner, *A conditional entropy bound for a pair of discrete random variables*, IEEE Trans. Inf. Theory, 21 (1975), pp. 493–501.

[113] S. Yang and R. Koetter, *Network coding over a noisy relay: a belief propagation approach*, in Proc. IEEE Int. Symp. Inf. Theory (ISIT), June 2007, pp. 801–804.

[114] Y. Yang, H. Hu, J. Xu, and G. Mao, *Relay technologies for WiMax and LTE-advanced mobile systems*, IEEE Commun. Mag., 47 (2009), pp. 100–105.

[115] S. Yao and M. Skoglund, *Analog network coding mappings in Gaussian multiple-access relay channels*, IEEE Trans. Commun., 58 (2010), pp. 1973–1983.

[116] R. Zamir, *Lattices are everywhere*, in Proc. Inf. Theory and Applications Workshop (ITA), Feb. 2009, pp. 392–421.

[117] G. Zeitler, G. Bauch, and J. Widmer, *Quantize-and-forward schemes for the orthogonal multiple-access relay channel*, IEEE Trans. Commun., 60 (2012), pp. 1148–1158.

[118] G. Zeitler, R. Koetter, G. Bauch, and J. Widmer, *Design of network coding functions in multihop relay networks*, in Proc. 5th Int. Symp. Turbo Codes and Related Topics, Sept. 2008, pp. 249–254.

[119] G. Zeitler, R. Koetter, G. Bauch, and J. Widmer, *On quantizer design for soft values in the multiple-access relay channel*, in Proc. IEEE Int. Conf. Commun. (ICC), June 2009.

[120] G. Zeitler, A. Singer, and G. Kramer, *Low-precision A/D conversion for maximum information rate in channels with memory*, IEEE Trans. Commun., 60 (2012), pp. 2511–2521.

[121] S. Zhang, S. C. Liew, and P. P. Lam, *Hot topic: Physical-layer network coding*, in Proc. 12th Annu. Int. Conf. Mobile Computing and Networking (MobiCom), Sept. 2006, pp. 358–365.