Diss. ETH No. 14760

Computing Information Rates of Finite-State Models with Application to Magnetic Recording

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Doctor of Technical Sciences

presented by
DIETER M. ARNOLD

born 10. April 1971
citizen of Seedorf (UR) and Luzern (LU)
dipl. El.-Ing. ETH

accepted on the recommendation of
Prof. Dr. Hans-Andrea Loeliger, examiner
Dr. Evangelos S. Eleftheriou, co-examiner
Prof. Dr. Aleksandar Kavčić, co-examiner

2003

Für meine Eltern

Acknowledgments

Above all, I would like to express my deepest gratitude to Andi Loeliger of the Swiss Federal Institute of Technology (ETH). I am obliged to my Doktorvater for making it possible, that I could work on such a rewarding project. He has been an ever sparkling fountain of ideas and enthusiasm. During the many pleasant sessions at the white-board, he taught me the importance of clear thinking leading to good solutions, characterized by simplicity and beauty.

Special thanks go to Evangelos Eleftheriou of IBM for giving freely from his expertise in magnetic recording and for making me feel part of his research group at the Zurich Research Laboratory (ZRL). Even in times of hectic day to day business, his door was always open for me.

Further, I am indebted to Alek Kavčić from Harvard University for his continued support and encouragement. His prompt and elaborate answers to all my questions at any time of the day made me wonder whether there is a six hour time difference between Zurich and Boston. In particular, I would like to thank Alek for arranging my stay at Harvard University, introducing me into his research group, and for his warm hospitality.

This thesis greatly benefited from the generosity and humor of my friends and colleagues. I would like to thank Pascal for the various adventures I could experience with him on our travels related to information theory. Further, I am indebted to Pascal for substantial contributions to my thesis and for sharing with me many long evenings at ETH.
I was lucky to have Xiaoyu as my officemate at IBM. He endured my outbursts of rage during endless debugging sessions with the imperturbable Asian friendliness. I would also like to thank Xiaoyu for sharing with me his ideas on low-density parity-check codes, and for introducing me into the beautiful world of Chinese home cooking.

Finally, I would like to thank Roy, Ajay, and Thomas, all members of the magnetic recording group at ZRL, for inspiring discussions and constructive criticism. Last but not least, I am grateful to all the members of the Signal and Information Processing Laboratory (ISI) at ETH, in particular to the legendary ISI-tigers.

Abstract

The topics of this thesis are the mathematical models of the magnetic recording channel and their ultimate information-theoretic limit, the capacity.

Source and channel of a magnetic recording system can be represented by a single finite-state model (FSM). The joint source/channel FSM is fully specified by the state-transition probabilities of the source model and the output probability distribution. In our case, this distribution is a parameterized Gaussian mixture density. We focus on aperiodic and irreducible FSMs whose state and observation process are stationary and ergodic. Thus, by the Shannon-McMillan-Breiman theorem, the entropy rates of the state and the observation process are determined by the probability of a typical sequence of those processes.

A new and practical method is presented for computing estimates of lower and upper bounds (information rates) on the capacity of FSMs. The pivotal observation behind the method is that the entropy rate of the channel output can be computed by standard forward sum-product trellis processing of simulated or (in principle) measured channel output data.

The method is applied to various FSMs representing the magnetic recording channel. These models are (generalized) partial-response polynomials with additive white Gaussian noise (AWGN), sources with run-length limit constraints observed through AWGN, FSMs that are trained on synthetically generated waveforms (microtrack model) with continuous mixture noise including medium noise, and the binary jitter channel with discrete-valued data-dependent noise.

Keywords: Channel capacity, finite-state models, information rate, magnetic recording, medium noise, Shannon-McMillan-Breiman theorem, sum-product algorithm.

Kurzfassung

Die Arbeit handelt von den mathematischen Modellen des Schreib- und Lesekanals in Systemen zur magnetischen Datenaufzeichnung und deren informationstheoretischen Grenze, der Kanalkapazität.

Bei der magnetischen Datenspeicherung kann der Schreib- und Lesevorgang durch ein einziges Modell mit endlicher Anzahl Zuständen (MEZ) dargestellt werden. Das MEZ für Quelle und Übertragungskanal ist durch die Verzweigungswahrscheinlichkeiten der Quelle und die Wahrscheinlichkeitsverteilung am Ausgang vollständig bestimmt. Wir nehmen als Verteilung eine parametrisierte Mischung von Gaussdichten an. Ferner betrachten wir aperiodische und nicht-zerlegbare MEZe, deren Zustands- und Ausgangsprozess stationär und ergodisch sind. Es gilt daher der Satz von Shannon-McMillan-Breiman, der besagt, dass die Entropieraten des Zustands- und Beobachtungsprozesses durch die Wahrscheinlichkeit einer typischen Sequenz dieser Prozesse bestimmt sind.

Eine neue und einfach anzuwendende Methode zur Berechnung von Schätzwerten der Informationsrate von MEZe wird vorgestellt. Diese Informationsraten stellen untere und obere Schranken für die Kanalkapazität dieser Modelle dar. Der Methode liegt die Erkenntnis zu Grunde, dass die Entropierate mittels eines einzigen Vorwärtslaufes des Summe-Produkt-Algorithmus berechnet werden kann. Dazu können simulierte oder (im Prinzip) gemessene Kanalausgangsdaten verwendet werden.

Verschiedene MEZe für den Schreib- und Lesekanals in Systemen der magnetischen Datenaufzeichnung werden mit dieser Methode studiert. Diese Modelle sind verallgemeinerte Polynome mit endlicher Stossantwort und AWGN, restringierte Markovquellen mit AWGN, mittels künstlich erzeugter Daten (Microtrack-Model als Kanalmodel) trainierte MEZe mit gemischtem Rauschen (AWGN und datenabhängigem Rauschen) und schliesslich der binäre Jitterkanal mit diskretem, datenabhängigem Rauschen.

Stichworte: Kanalkapazität, Modelle mit endlichem Zustandsraum, Informationsrate, magnetische Datenaufzeichnung, datenabhängiges Rauschen, der Satz von Shannon-McMillan-

Breiman, Summe-Produkt-Algorithmus.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

RAMAC (which stands for random access method of accounting and control) was the world's first computer disk storage system, developed by IBM engineers in San Jose, CA, and introduced in 1957. Prior to this, computer storage was largely reliant on magnetic tape. The disk-based storage introduced movable read/write heads, which enabled a semi-random access capability; this ability was a momentous achievement, both for IBM and for the computing world in general, because fast random access to large volumes of data made interactive computer systems practical.

Although less popular than the celebrated chip technology, disk-based storage is considered a major driver of the information technology (IT) revolution in the past four decades. The most important customer-oriented parameter of disk storage is the cost per megabyte. Lower costs are achieved by packing more and more bits on the same surface area, resulting in what is termed increase in "areal density".

Areal density in magnetic recording increased from 100 MBits/in^2 in 1991 to 18 GBits/in^2 in 2001, i.e. a 180-fold improvement. In the same period, the speed of personal computers augmented from 40 MHz to 1.5 GHz — a 40-fold increase in speed only. Thus, the areal density road map of magnetic recording outpaced the chip road map, known as Moore's law, by a factor of 4, and it can truly be said that disk-based storage is the "unsung" hero of IT!

This thesis is about the mathematical models of the magnetic recording channel and their ultimate information-theoretic limit, the capacity. We present a practical method for computing bounds on the capacity of such channel models and thereby reveal potential gains in areal density in today's products.

1.1 Motivation

Consider the binary-input linear channel with memory two and additive white Gaussian noise (AWGN) at the output depicted in Fig. 1.1. It serves as a model for the data channel in hard disk-drives. As simple as this channel model looks, its information-theoretic limit, the channel capacity, is unknown. In fact, the capacity computation of intersymbol interference (ISI) channels with discrete input has remained an open problem in information theory since its inception by C. E. Shannon in 1948. Surprisingly, not even a practical algorithm has been known for the much simpler problem of computing the average mutual information between the input and the output of such ISI channels for the case that the discrete input is independent and uniformly distributed (i.u.d.) over the input alphabet. This problem has usually been circumvented by neglecting the channel memory, leading to loose bounds on capacity. On the occasion of the 50th anniversary of information theory, Immink, Siegel, and Wolf state in their contribution [46] to the commemorative issue that: *The general problem of computing, or developing improved bounds for, the capacity of discrete-time ISI models of recording channels remains a significant challenge.*

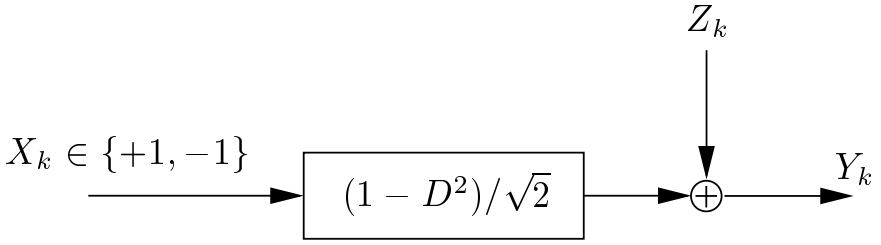

Figure 1.1: ISI channel with discrete input.

In addition to being an exciting theoretical challenge, the computation of tight bounds on capacity has also substantial practical implications. In the past, major breakthroughs in the head and material technologies have been responsible for the spectacular growth in storage density. But now, signal processing and coding are recognized as cost-efficient means to further improve areal density. The ultimate limit of any signal processing and coding scheme is the channel capacity. The signal-to-noise ratio (SNR) gap between the capacity and the performance — in terms of sector error-rate — in today's products can (in principle) be closed by more sophisticated signal-processing and coding schemes. Thus, knowing the capacity reveals how much current signal processing and coding schemes can be improved. This improvement would result in the same sector error-rate performance at a lower SNR, which implies that more bits can be packed on the same surface, i.e. an increase in areal density.

1.2 Survey of the Literature

The information rate between the input process $\mathcal{X} = (X_1, X_2, \ldots)$ and the output process $\mathcal{Y} = (Y_1, Y_2, \ldots)$ of a discrete-time channel with memory m and some well-defined starting state s_0 is given in the limit as

$$I(\mathcal{X};\mathcal{Y}|S_0 = s_0) \triangleq \lim_{n\to\infty} \frac{1}{n} I(X_1, \ldots, X_n; Y_1, \ldots, Y_n|S_0 = s_0). \tag{1.1}$$

In general, this limit, i.e. the information rate, may not be well-defined and/or may depend on the starting state s_0. We will assume that $\mathcal{X}$ is Markov or hidden Markov, and we are primarily interested in the case where the channel input alphabet $\mathcal{X}$ (i.e., the set of possible values of X_k) is finite.

In [34], Gallager defined finite-state channels (FSC) and introduced the notion of *indecomposable* FSCs. The information rate of indecomposable FSCs is well-defined and does not depend on the starting state. Nevertheless, in many cases of practical interest, the computation of (1.1) is a problem. Analytical simplifications of (1.1) are usually not available even if the input symbols X_k are i.u.d.. The complexity of the direct numerical computation of

$$I(n) \triangleq \frac{1}{n} I(X_1, \ldots, X_n; Y_1, \ldots, Y_n|S_0 = s_0) \tag{1.2}$$

is exponential in n, but the sequence $I(1), I(2), I(3), \ldots$ converges rather slowly even for channels with small memory.

The problem of computing the information rate for i.u.d. input was studied by Hirt [41] and by Shamai et al. [68]. Using input blocks of finite length n, Hirt [41] evaluated directly (1.1) and obtained via stochastic averaging (Monte-Carlo simulations) estimates of an upper and a conjectured lower bound on the i.u.d. information rate. The upper bound is not an upper bound on the information rate but rather on the conjectured lower bound. If m is the channel memory, the difference between the upper bound and the conjectured lower bound is at most m/n bits/symbol, and the complexity of this method is proportional to 2^n. The bounds reported in [68] neglect the memory in the channel and are loose even for channels with small memory.

Mushkin and Bar-David [57] analyzed the Gilbert-Elliot channel and Goldsmith and Varaiya [37] extended that work to a more general channel model with a freely evolving state. For such a channel model, the channel state is independent of the input. Thus, the channel model used in [37] does not include ISI channels. Goldsmith and Varaiya gave expressions for the information rate and the capacity of their channel model and showed that both depend on the stationary distribution of the channel state. Exploiting the Markov structure of the channel,

Goldsmith and Varaiya devised recursive methods for the evaluation of the stationary state-distribution as well as the information rate and the capacity of channels with a freely evolving state.

Recently, a lower bound on the capacity has been conjectured by Shamai and Laroia [67]. In contrast to Hirt's method, this conjectured lower bound is easy to compute even for channels with large memory. By numerical comparison, it has been observed [67] that the conjectured Shamai-Laroia lower bound (CSLLB) on the capacity lies in the region of Hirt's bound on the i.u.d. information rate for a channel with memory one. The tightness of the CSLLB depends on the channel memory and the relative values of the coefficients of the channel transfer function. For a channel with memory six and Gaussian-like distributed coefficients, we found that the inherent assumption of the CSLLB, i.e. that the mixture of ISI and AWGN is Gaussian distributed, is well met. In [5], it was shown that numerical results obtained from the CSLLB coincide with the ones obtained from the forward sum-product method — the method about which actually this thesis is. In addition to being easily computable, the CSLLB is attractive, because it is based on the power spectral density of the underlying channel. Thus, the influence of various channel parameters on the information rate is transparent. As a substitute for the channel capacity, the CSLLB was applied in [44] to partial-response class-4 polynomials and in [26] to generalized partial-response polynomials to assess different coding schemes. In [3], the CSLLB was applied to the microtrack channel model to investigate the effects of medium noise in magnetic recording systems on the channel capacity. As the CSLLB requires the underlying channel to be linear, and as channels with data-dependent (medium) noise are nonlinear, this was only possible by considering the average channel transfer function — averaged over all possible input patterns. In [2], the CSLLB was generalized to Markovian input processes, enabling the computation of non-i.u.d. information rates.
However, the CSLLB possesses several disadvantages: first it is a *conjectured* lower bound. Thus all results have to be taken with caution. Moreover, no upper bound is known that bases on the same method. As the CSLLB requires the underlying channel to be linear, it can only be applied to data-dependent channels using an average power spectral density. Finally, the closely related problem of computing information rates of constraint input sources, such as run-length-limited sources, observed through a memoryless noisy channel cannot be addressed.

The capacity of linear continuous-time channels with peak constraint input and AWGN at the output was investigated by Shamai and Bar-David [64, 65]. In [65], the input signal was assumed to be a continuous-time stationary binary signal with zero mean that can take on only the values $+1$ and -1 with equal probability. If no restriction is made on the distribution of the transition from -1 to $+1$ and vice-versa, the binary input process is termed a *unit process*. It encompasses as special case the random telegraph waveform, where the transitions

are distributed according to a Poisson distribution. Unit processes were studied by Shepp [72] who proved the covariance description of unit processes discovered by McMillan in the early 50s [56]. McMillan showed that a unit process must fulfill not only an infinite but also an uncountable number of constraints (see also [55]).

For such an unit process, Wu et al. computed upper bounds on the capacity of the magnetic recording channel with stationary medium noise [78]. They used the covariance description and frequency interpretation of unit processes together with Gaussian input signals and the "water-filling" technique to compute upper bounds on the capacity of the constraint channel. The starting point of the method presented in [78] is a finite number of constraints such that the optimization is easily executed by the "water-filling" technique. By progressively increasing the number of constraints, the upper bound becomes tighter and the number of allowed input sequences is constantly thinned out such that at the end (ideally) only those sequences remain which fulfill the ± 1 constraint. The weakness of this approach is that in practice only a small fraction of all constraints can be considered, that it cannot be used for data-dependent, nonlinear channels, and that it does not provide lower bounds on the capacity.

French and Wolf computed upper and conjectured lower bounds on the capacity for the magnetic recording channel for various noise scenarios (including medium noise) by assuming Gaussian inputs and physically motivated channel models [33]. However, the Gaussian assumption fails — in particular at high rates, where our main interest resides. Moreover, the channel models of French and Wolf are not easily describable and are difficult to use for signal processing and coding.

The effect of pulse jitter on the capacity of binary input ISI channels was investigated using the Arimoto–Blahut algorithm in [58]. To this end, the channel output was quantized to three levels resulting in considerable quantization loss even for the binary phase-shift keying channel and AWGN, i.e. without memory and jitter.

The capacity of run-length limited (RLL) sequences over noisy channels is a closely related problem and has been investigated by Zehavi and Wolf [79], Shamai et al. [66], and Heegard et al. [39, 40]. Zehavi and Wolf considered RLL sequences transmitted over the discrete-time binary symmetric channel (BSC). They derived a set of lower bounds theoretically by assuming the input to be Markov, and showed results obtained by brute-force computation for the case where the order of the Markov input process equals the minimal state-space realization of the RLL constraint. Shamai et al. presented in [66] upper and lower bounds on the information rate of noisy RLL sequences for the BSC as well as the AWGN channel by exploiting the property of stationarity. Heegard et al. focused on the continuous-time AWGN channel with peak constraint input [39, 40]. The bounds are quite loose.

1.3 Objective of the Thesis

The objective of this thesis is to present a practical method for computing estimates of information rates of finite-state channel models. The method consists of two steps: a collection and a computation step. First, a large amount of data is collected from the channel model, which can be done either by simulations or (in principle) by measurements. The collected data comprise the entire statistics of the channel model. In a second step, the data are processed on the trellis induced by the joint source/channel model. The pivotal observation behind the method is that estimates of entropy rates and, hence, information rates of finite-state channel models can be computed exactly by standard forward sum-product trellis processing of simulated or (in principle) measured data.

Three key properties of FSMs make this possible:

- *Simple representation of FSMs:* An FSM is fully specified by the state-transition probabilities and the parameterized conditional output distribution.

- *Stationarity and ergodicity:* The output process of a stationary and ergodic FSM is stationary and ergodic. Thus, by the Shannon-McMillan-Breiman theorem, the entropy rate of the output process is determined by the probability of a typical sequence.

- *Markov property:* The Markov structure of the FSM allows to factor the probability measure of the state as well as the output process and enables efficient computation methods.

The main contribution of this thesis is not a theoretical result but a practical computation method. Remembering R. W. Hamming's maxim that *The purpose of computing is insight, not numbers* and A. Ralston's reply that *...but for the student, numbers are often the best road to insight* this method is underpinned by theoretical results as well as elucidated by various computer experiments. The application in mind here is magnetic recording in longitudinal direction, although the presented method clearly is not restricted thereto.

1.4 Outline of the Thesis

The thesis is subdivided into five chapters and three appendices. The first chapter contains the introduction.

Background and Definitions

The second chapter introduces the mathematical notation and terminology along with the three cornerstones on which the thesis builds: finite-state models, magnetic recording, and information theory. The subsection on finite-state models and information theory state well-known results and build up the mathematical framework needed for the main part of the thesis, Chapter 3. The subsection on magnetic recording starts with briefly explaining the magnetization process, continues with the ideal write and read process, and leads to the magnetic recording channel models used in today's hard-disk drives, i.e. the partial-response channel models. The chapter concludes with the problem statement.

Sampling-based Computation of Information Rates

The third chapter contains the main part of the thesis. It presents a new method for computing estimates of entropy rates of hidden Markov models. This method can be considered as a stochastic version of the "power-method" which is commonly used to compute the largest eigenvalue and corresponding largest eigenvector of non-negative primitive matrices. The proposed method can be used to exactly compute estimates of information rates of finite-state source/channel models that are ubiquitous in magnetic recording — but arise in other areas as well. Moreover, algorithms are presented for shaping the Markov source to the channel. Such Markov sources lead to noticeably higher information rates and thus to tight lower bounds on the channel capacity. Further, a new upper bound on the capacity of finite-state channels is proposed. This upper bound is obtained in a similar spirit as the lower bound. Finally, the sampling-based computation method is extended to compute upper and lower bounds on the information rate of general (non-finite state) ergodic channels. The chapter concludes with reduced-state versions of these bounds.

Information Rates of Magnetic Recording Channel Models

Numerical results are provided for various channel models. The first part focuses on channel models without medium noise. For selected channel models, estimates of information rates are presented under the assumption that the input is i.u.d.. Next, estimated information rates are presented that result from optimizing the state-transition probabilities of the Markov input source. It is shown that by increasing the order of the input source over its minimal state-space realization, a set of tight lower bounds on the channel capacity is obtained. Numerical results for the proposed upper bound are provided as well.
The second part of this chapter is devoted to channel models for medium noise. Such data-

dependent channel models are obtained by training a parameterized FSM with synthetically generated waveforms (microtrack model). By means of such an auxiliary FSM, the information rate of the microtrack channel can be lower bounded. Finally, inspired by the microtrack channel, a new model for the medium-noise-dominated magnetic recording channel is presented along with numerical results: the binary jitter channel.

Summary and Concluding Remarks

The last chapter summarizes the thesis and states some concluding remarks.

Appendices

Appendix A contains a numerical example of a Markov source (a $(0, 2)$-RLL constrained Markov source) to illustrate the concepts presented in Chapter 2.
Appendix B presents the well-known Perron-Frobenius theorem from matrix theory and details the so-called "power-method". It touches the heart of this thesis, namely the interwoven connection between analytical (combinatorial) and sampling-based (probabilistic) computation of entropy rates of FSMs.
In Appendix C, Shannon's formula for maxentropic, i.e. capacity-achieving, state-transition probabilities of Markov sources is derived.

Chapter 2

Background and Definitions

The three cornerstones of this thesis are finite-state models, magnetic recording, and information theory. Finite-state models serve as source and channel model for magnetic recording systems. They are related to information theory via stationarity and ergodicity, i.e. the Shannon-McMillan-Breiman theorem. This allows us to use finite-state models as a computation vehicle for accurately estimating information rates of magnetic recording systems.

2.1 Finite-State Models

A finite-state model $\mathcal{M}_{\boldsymbol{\theta}(L)}$ is a doubly stochastic random process. It consists of a non-observable process $\mathcal{S}$, called state process, and an observable process $\mathcal{Y}$, called output process. The former is of finite size, i.e. $L = |\mathsf{S}| < \infty$, and determines the *structure* of the finite-state model. The latter, the observable output process, can take on values in a finite or infinite alphabet. It can be a deterministic or probabilistic function of the state process, and it is characterized by a parameter vector $\boldsymbol{\theta}(L)$ which belongs to $\boldsymbol{\Theta}(L)$, a subset of the Euclidean space with dimension Υ (which is proportional to L). The output process inherits its statistical properties from the underlying state process.

The stochastic state process is assumed to be a Markov process. The state-space of a Markov process contains a countable (possibly infinite) number of states [30, 38]. Hidden Markov models (HMMs) form a large and useful class of stochastic process models. They assume a sequence of random variables to be conditionally independent given a sequence of state variables which forms a Markov process. In signal processing and estimation problems, HMMs are usually assumed to have a finite state-space [10–13, 62], although they can be extended to infinite state-spaces [15]. We will use the notion of a finite-state model (FSM) for

a HMM with *finite* state-space. The observable output process of the FSM is termed *hidden Markov process* (HMP) [28].

2.1.1 Structure

States and State-Transitions

The state process of an FSM is determined by states and branches connecting the states. More formally: Let the *state-space* S be a nonempty finite set with cardinality $|\mathsf{S}| < \infty$, the elements of which will be called *states*. The cardinality of the state set, $L = |\mathsf{S}|$, is termed structure index or *order* of the FSM. Let B be a finite set, the elements of which will be called *state-transitions* or *branches*. Every branch $b \in \mathsf{B}$ has a well-defined *left state* $\mathrm{lst}(b) \in \mathsf{S}$ and a well-defined *right state* $\mathrm{rst}(b) \in \mathsf{S}$. Two branches b and b' are *parallel* if both $\mathrm{lst}(b) = \mathrm{lst}(b')$ and $\mathrm{rst}(b) = \mathrm{rst}(b')$. We will assume that both maps $\mathrm{lst}(\cdot) : \mathsf{B} \to \mathsf{S}$ and $\mathrm{rst}(\cdot) : \mathsf{B} \to \mathsf{S}$ are surjective (i.e., there are no unused states).
A *path* of length n in an FSM is a sequence $b^n = (b_1, b_2, \ldots, b_n)$ of branches $b_k \in \mathsf{B}$, $k = 1, 2, \ldots, n$, such that $\mathrm{rst}(b_k) = \mathrm{lst}(b_{k+1})$. Note that every path b^n originates from a unique state sequence, i.e. $s_0^n = (s_0, s_1, \ldots, s_n) = \big(\mathrm{lst}(b_1), \mathrm{rst}(b_1) = \mathrm{lst}(b_2), \ldots, \mathrm{rst}(b_n)\big)$.

Adjacency Matrix, Directed Graph, and Trellis Representation

The structure of an FSM can be represented by the *adjacency matrix* $\mathbf{A}$ of size $L \times L$. Each entry $\mathbf{A}(i,j) = [\mathbf{A}]_{ij}$ in row i and column j for $i, j \in \mathsf{S} = \{1, \ldots, L\}$ of $\mathbf{A}$ is assigned a 1 if state i is directly (without any intermediate state) connected to state j. If there is no such branch, that entry is a 0. Clearly, matrix $\mathbf{A}$ is non-negative, denoted as $\mathbf{A} \geq \mathbf{0}$. Note that we preclude parallel branches for the sake of simplicity and without loss of generality of the subsequent results[1].
The structure of the FSM is visualized in a directed graph, termed *state-transition diagram*, where each branch is labeled with the associated output symbol. Unfolding the state-transition diagram over time results in a time-invariant *trellis* representation.
A trellis is defined as follows. Let n be a positive integer. There is a collection $\mathsf{S}_0, \mathsf{S}_1, \ldots, \mathsf{S}_n$ of nonempty disjoint finite sets; the elements of S_k are called time-k states. There is also a collection $\mathsf{B}_1, \mathsf{B}_2, \ldots, \mathsf{B}_n$ of nonempty disjoint finite sets; the elements of B_k are called time-k branches. Each branch $b_k \in \mathsf{B}_k$ has a well-defined left state $\mathrm{lst}(b_k) \in \mathsf{S}_{k-1}$ and a well-defined right state $\mathrm{rst}(b_k) \in \mathsf{S}_k$. We will assume that $\mathrm{lst}(\mathsf{B}_k) = \mathsf{S}_{k-1}$ and $\mathrm{rst}(\mathsf{B}_k) = \mathsf{S}_k$.
One also says that a trellis of length n consists of n concatenated *trellis sections*. The time-k trellis section T_k is a set that contains all *possible* transitions $(s_{k-1}, s_k) = (i, j)$, i.e. $\mathsf{T}_k = \{(i,j) \in \mathsf{S}_{k-1} \times \mathsf{S}_k | \mathbf{A}(i,j) = 1\}$. If the trellis is time-invariant, all trellis sections are identical,

[1] An example for an FSM with parallel branches is given in Appendix A.

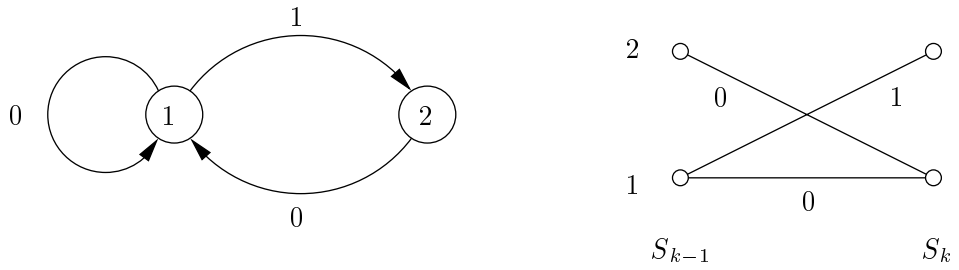

Figure 2.1: State-transition diagram and trellis section of a $(1,\infty)$-RLL source.

and the time index k is neglected.

Example 2.1 ($(1,\infty)$-RLL Source)
Consider a binary source emitting 0s and 1s in such a way that each 1 must be followed by at least one 0. Such a source is termed a $(d=1,k=\infty)$-RLL source because the minimum number of 0s between two 1s must be $d=1$, and the maximum number of 0s between two 1s is not restricted, i.e. $k=\infty$ (for a definition of (d,k)-RLL sources see Appendix A).
A state-transition diagram and the corresponding trellis section are shown in Fig. 2.1. The branches are labeled with the associated output symbols. We observe that the (d,k)-constraint can be satisfied with two states, i.e. $L=2$, and three branches. The corresponding 2×2 adjacency matrix contains three 1s, each 1 representing one branch, i.e.

$$\mathbf{A}=\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}.$$

Example 2.2 (DICODE Channel)
Consider a linear discrete-time channel with impulse response $(1-D)/\sqrt{2}$. Assume that the time-k input X_k is constrained to be bipolar, *i.e. $x_k\in\{+1,-1\}$. This channel is termed DICODE channel in magnetic recording. The time-$(k-1)$ state is given by the time-$(k-1)$ bipolar input in the following way $S_{k-1}=0.5\cdot(X_{k-1}+3)$. The time-$k$ output $V_k=(X_k-X_{k-1})/\sqrt{2}$ with $X_0=-1$ can take on values in the set $\mathsf{V}=\{-\sqrt{2},0,+\sqrt{2}\}$. A state-transition diagram and the corresponding trellis diagram are depicted in Fig. 2.2 with the associated input and output pair x_k/v_k on each branch. As all states are connected with each other, the corresponding adjacency matrix is given by*

$$\mathbf{A}=\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}.$$

For any integer $n>1$ the (i,j)-th entry of the n-th power of $\mathbf{A}$ equals the number of all possible paths of length n starting in state i and ending in state j after n time steps, i.e.

$$[\mathbf{A}^n]_{ij}=\sum_{k_1,k_2,\cdots,k_{n-1}}\mathbf{A}(i,k_1)\cdot\mathbf{A}(k_1,k_2)\cdots\mathbf{A}(k_{n-1},j) \tag{2.1}$$

where the summand is zero unless the state sequence $s_0^n=(i,k_1,\ldots,k_{n-1},j)$ induces a valid path $b^n=(b_1,b_2,\ldots,b_n)$, i.e. unless

$$\big(i=\mathrm{lst}(b_1)\big)\cap\big(\mathrm{rst}(b_1)=k_1\big)\cap\cdots\cap\big(\mathrm{rst}(b_n)=j\big). \tag{2.2}$$

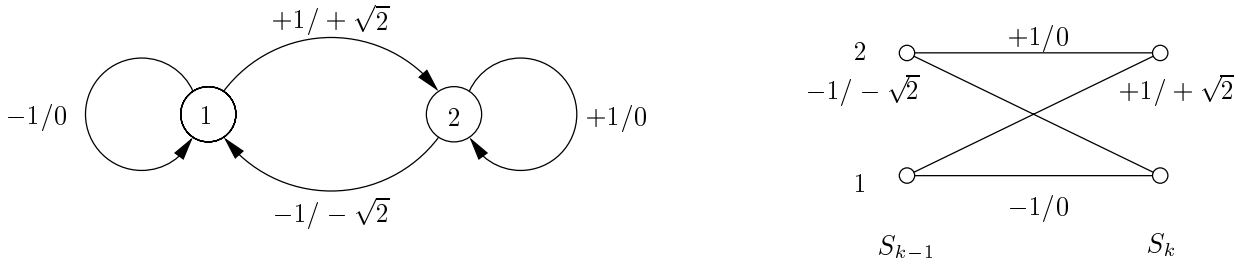

Figure 2.2: State-transition diagram and trellis section of the DICODE channel.

If there exits more than one path, the summand equals the number of all such paths.

Example 2.3 ($(1,\infty)$-RLL Source)
There are two paths in the $(1,\infty)$-RLL source that lead from state 1 to state 1 in two time steps (see Fig. 2.3 left part), i.e.

$$\mathbf{A}^2 = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}.$$

The adjacency matrix $\mathbf{A}$ implicitly assumes a direction along the time-axis from left to right. To be precise, we should call $\mathbf{A}$ the *forward* adjacency matrix $\mathbf{A}_\mathrm{f}$, but, as we will generally operate in the forward direction, we will not do so. The *backward* adjacency matrix $\mathbf{A}_\mathrm{b}$ is the transpose of the forward adjacency matrix (see Appendix A for an illustrative example).

Moreover, we assume that the adjacency matrix $\mathbf{A}$ contains all state-transitions that are possible within *one* time step (as outlined above). Hence, we do not call $\mathbf{A}^2$ an adjacency matrix, because it contains all state-transitions within two time steps. This restriction is somewhat arbitrary and does not change the subsequent results (see also Appendix A), but it is a consequence of our assumption that the adjacency matrix contains no parallel branches.

Minimal State-Space Realization

The $(1,\infty)$-RLL constraint can be realized with different FSMs. In Fig. 2.1, a two-state FSM is depicted. A three-state realization is shown in Fig. 2.3 (right part). Of course, the two-state realization is computationally less expensive and therefore, in the light of trellis-processing algorithms (such as the Viterbi or the sum-product algorithm), preferable in general.
We are interested in a *minimal state-space realization*, i.e. an FSM with the smallest number of states (or equivalently the smallest order) such that all constraints are fulfilled. A minimal state-space realization is in general not unique, because the states are not strictly identifiable [52].

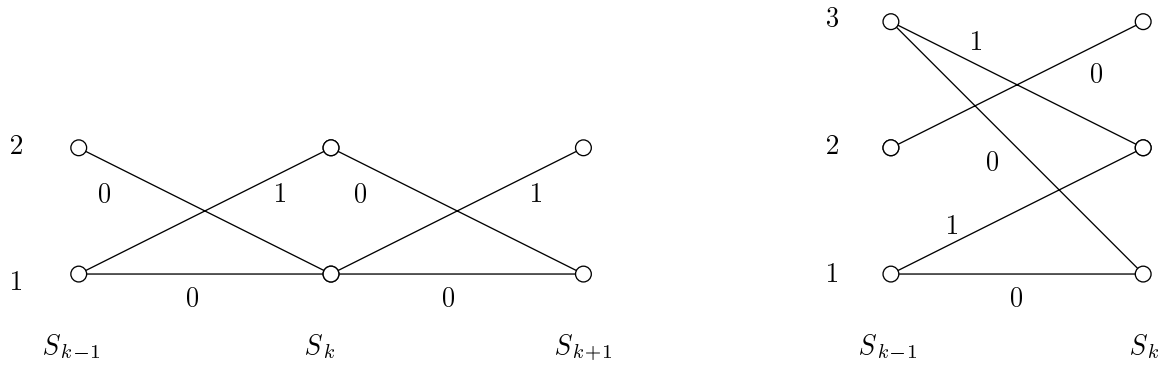

Figure 2.3: Two trellis sections of the $(1,\infty)$-RLL trellis (left part) and a realization of the $(1,\infty)$-RLL constraint with three states (right part).

We will not address the question of how to find the minimal state-space realization for a given set of constraints, but assume to have one available. Note that the minimal state-space realization does not mean that the adjacency matrix has full rank. A counterexample is the DICODE channel.

2.1.2 Markov Property

We assume that the sequence of states S_k forms a *Markov process* in the sense that the probability of being in state j at time k conditioned on all states up to the state i at time $k-1$, depends only on the state i at time $k-1$. Thus, given the present state, the future is independent of the past. More precisely, we have

$$Pr(S_k = j | S_{k-1} = i, S_{k-2} = i', \ldots) = Pr(S_k = j | S_{k-1} = i), \tag{2.3}$$

which is called the *Markov property*. The probability of going from state $S_{k-1} = i$ to state $S_k = j$ is called *forward state-transition probability* (STP). The STPs are collected in the *forward state-transition probability matrix* $\mathbf{Q}$ of size $L \times L$, where the entry $\mathbf{Q}(i,j)$ in row i and column j equals the STP, i.e.

$$\mathbf{Q}(i,j) \triangleq Pr(S_k = j | S_{k-1} = i). \tag{2.4}$$

For each row, the row entries must sum up to one, i.e.

$$\sum_{j \in \mathsf{S}} \mathbf{Q}(i,j) = \sum_{j \in \mathsf{S}} Pr(S_k = j | S_{k-1} = i) = 1 \qquad \forall i \in \mathsf{S}. \tag{2.5}$$

Thus, the matrix $\mathbf{Q}$ is termed a *stochastic* matrix and contains positive entries only at positions where the adjacency matrix $\mathbf{A}$ has positive entries, i.e. $\mathbf{Q}$ is non-negative as well.

Example 2.4 ($(1,\infty)$-RLL Source)

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}(1,1) & \mathbf{Q}(1,2) \\ \mathbf{Q}(2,1) & 0 \end{pmatrix}$$

with $\mathbf{Q}(1,1) + \mathbf{Q}(1,2) = 1$ *and* $\mathbf{Q}(2,1) = 1$.

There exists also a *backward state-transition probability matrix* $\mathbf{Q}_\mathrm{b}$ with entries $\mathbf{Q}_\mathrm{b}(i,j) = Pr(S_{k-1} = j | S_k = i)$, i.e. positive entries only at positions where $\mathbf{A}_\mathrm{b}$ contains positive entries. Thus, $\mathbf{Q}_\mathrm{b}$ is a stochastic and non-negative matrix as well. Note that $\mathbf{Q}_\mathrm{b}$ cannot be obtained simply by transposing the forward state-transition probability matrix $\mathbf{Q}$, as the row constraint is the same for both directions (forward and backward). In other words, while the time axis can be changed from forward to backward direction through transposition, i.e. $\mathbf{A}_\mathrm{b} = \mathbf{A}^T$, the forward and backward state-transition probability matrices, $\mathbf{Q}$ and $\mathbf{Q}_\mathrm{b}$, are both subject to a row constraint (for an illustrative example see in Appendix A).

In general, the state-transition probabilities of a Markov source may depend on time. We do not assume this here and therefore say that the Markov process is *homogenous* in time. Such Markov processes are called *Markov chains* [30].

2.1.3 Persistent, Aperiodic, and Irreducible Markov Chains

A state i of a Markov chain is said to be *persistent* if starting in state i the chain is certain sometime to return to state i, i.e. if

$$Pr\left(\bigcup_{k=1}^{\infty} [S_k = i] | S_0 = i\right) = 1. \tag{2.6}$$

A Markov chain is presistent, if all states of the Markov chain are presistent.

Consider a Markov chain with a finite state-space. A state i is called *periodic* with period $d > 1$, if the probability of re-entering state i in n steps is zero unless n is divisible by d, and d is the largest integer with this property. A state with period $d = 1$ is called *aperiodic*. A Markov chain is said to be aperiodic if all its states have period 1.

A state i of a Markov chain is said to be accessible from state j if there is a finite sequence of transitions from j to i with positive probability. If the states i and j are accessible to each other, the states i and j of the Markov chain are said to communicate with each other and belong to the same *class*. In general, a Markov chain consists of one or more disjoint classes. If all the states belong to a single class, the Markov chain is referred to as *irreducible*. In other words, starting at any state of an irreducible Markov chain, we can reach any other state with

positive probability, i.e. for every pair (s, s') of states, there is a positive and finite integer n such that

$$Pr(S_{k+n} = s' | S_k = s) > 0. \tag{2.7}$$

If the Markov chain is finite (as we generally assume), (2.7) guarantees that all states of the Markov chain are persistent and the chain is then said to be persistent. Thus, for finite-state Markov chains the property of presistence follows from the property of irreducibility. Moreover, if the Markov chain is aperiodic, (2.7) guarantees that there exists a positive and finite integer n such that the matrix $\mathbf{A}^n$ is positive, denoted by $\mathbf{A}^n > \mathbf{0}$, i.e. $\mathbf{A}^n(i,j) > 0 \ \forall \ (i,j) \in \mathsf{S}^2$ (*mixing property*).

2.1.4 Stationary State-Distribution

Existence

Let the row vector $\boldsymbol{\pi}^{(k)}$ of length L denote the *state-distribution vector* of a Markov chain at time k. The i-th element of $\boldsymbol{\pi}^{(k)}$ is by definition the probability that the chain is in state i at time k, i.e.

$$\boldsymbol{\pi}^{(k)}(i) \triangleq Pr(S_k = i). \tag{2.8}$$

The state-distribution vector at time k is given by the state-distribution vector at time $k-1$ in the following way:

$$\boldsymbol{\pi}^{(k)} = \boldsymbol{\pi}^{(k-1)}\mathbf{Q}. \tag{2.9}$$

By iteration, we obtain

$$\boldsymbol{\pi}^{(k)} = \boldsymbol{\pi}^{(0)}\mathbf{Q}^k, \tag{2.10}$$

where $\boldsymbol{\pi}^{(0)}$ is the *initial* state-distribution vector. In words, the state-distribution vector of a finite-state Markov chain at time k is the product of the initial state-distribution vector $\boldsymbol{\pi}^{(0)}$ and the k-th power of the stochastic state-transition probability matrix $\mathbf{Q}$.
A Markov chain is said to be a stationary process if and only if it has a *stationary* state-distribution $\boldsymbol{\pi}$ such that $\boldsymbol{\pi}^{(k)} = \boldsymbol{\pi} \ \forall k$ or equivalently,

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{Q}. \tag{2.11}$$

Thus, the stationary state-distribution vector $\boldsymbol{\pi}$ is the left eigenvector corresponding to the largest eigenvalue (which is actually 1) of $\mathbf{Q}$. Note that (2.11) may in general not have a solution, and when it has one, it may not be unique.

Convergence to the Stationary Distribution

If the finite-state Markov chain $\mathcal{S}$ is irreducible, the stationary distribution is positive, i.e. $\boldsymbol{\pi}(s) > 0 \; \forall \; s \in \mathsf{S}$, and *unique* (see [28] and references therein). Thus, $\mathcal{S}$ is a stationary process. The question is now whether any initial state-distribution vector $\boldsymbol{\pi}^{(0)}$ converges to the stationary state-distribution vector $\boldsymbol{\pi}$ and if so how this convergence behaves.

For a finite-state, irreducible, and aperiodic Markov chain it holds that all states are ergodic [30], i.e.

$$\lim_{n\to\infty}[\mathbf{Q}^n]_{ij} = \boldsymbol{\pi}(j) \qquad \forall \;\; i,j \in \mathsf{S} \tag{2.12}$$

and the Markov chain is said to be an *ergodic process* [17]. From (2.10), it follows that for $n \to \infty$

$$\boldsymbol{\pi} = \boldsymbol{\pi}^{(0)}\mathbf{Q}^{\infty}, \tag{2.13}$$

i.e. that any initial state-distribution converges to the stationary state-distribution which is then called *steady-state distribution*. Moreover, the convergence occurs at an exponential rate [17], i.e.

$$|[\mathbf{Q}^n]_{ij} - \boldsymbol{\pi}(j)| \leq c\psi^n, \tag{2.14}$$

where $c \geq 0$ and $0 \leq \psi < 1$. This implies that there exists an integer n such that $[\mathbf{Q}^n]_{ij} > 0$ for all $i,j \in \mathsf{S}$ (mixing property of STPs). An ergodic Markov chain satisfying (2.14) is called *geometrically ergodic* [29].

A matrix $\mathbf{Q}$ is called *primitive*, if there is an integer n such that $\mathbf{Q}^n > \mathbf{0}$, i.e. all entries of the n-th power of $\mathbf{Q}$ are positive. The smallest such integer n is called the *index of primitivity* of $\mathbf{Q}$.
The state-transition probability matrix $\mathbf{Q}$ of a finite-state, irreducible, and aperiodic Markov chain is primitive [17]. A finite-state Markov chain with primitive $\mathbf{Q}$ possesses thus a unique positive stationary state-distribution. Moreover, this stationary state-distribution is a the steady-state distribution and the chain is an ergodic process.

Note that a sufficient, but not necessary condition, for a Markov chain to be *ergodic* is for it to be aperiodic.

Example 2.5 (Switching Process)
Consider a switching process given by the adjacency matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

This is a finite-state and irreducible process. Thus, the stationary state-distribution is unique and positive; in fact $\boldsymbol{\pi} = (0.5\ 0.5)$*. The process is therefore stationary. Moreover, the process is also ergodic, because the long-run relative frequency of a sequence (generated by this process) converges stochastically to the probability assigned to the sequence [6]. However, the mixing property of (2.12) is not satisfied and there is no steady-state distribution, i.e. the chain does not convergence to the stationary state-distribution (except if the initial state-distribution is the stationary state-distribution itself).*

2.1.5 Ergodicity Theorem and Representation of Markov Chains

There exists a vast literature on ergodic theorems for general Markov processes. As we focus on finite-state, irreducible, and aperiodic Markov chains, we summarize the important properties for such processes from [6, 28, 30, 38] in the following theorem.

Theorem 2.1 (Ergodicity Theorem for Finite-State, Irreducible, and Aperiodic Markov Chains) *Let a finite-state ergodic Markov chain with state sequence* $S_0, S_1, \ldots, S_n$ *and a stochastic state-transition matrix* $\mathbf{Q}$ *be irreducible and aperiodic. The chain has a unique stationary distribution to which it converges from any initial state. This stationary distribution* $\boldsymbol{\pi}$ *is called the steady-state distribution and it fulfills the following properties*

$$1. \quad \lim_{n\to\infty} [\mathbf{Q}^n]_{ij} = \boldsymbol{\pi}(j) \qquad \forall\ i \in \mathsf{S} \tag{2.15}$$

$$2. \quad \boldsymbol{\pi}(j) > 0 \qquad \forall\ j \in \mathsf{S} \tag{2.16}$$

$$3. \quad \sum_{j\in\mathsf{S}} \boldsymbol{\pi}(j) = 1 \tag{2.17}$$

$$4. \quad \boldsymbol{\pi}(j) = \sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i)\mathbf{Q}(i,j) \qquad \forall\ j \in \mathsf{S}. \tag{2.18}$$

Conversely, suppose that the finite-state Markov chain is irreducible and aperiodic, and that there exists a vector $\boldsymbol{\pi}$ satisfying conditions 2. – 4.. Then the chain is ergodic, and the steady-state probabilities are given by the first condition. Note that property 4. in the theorem above is termed *global balance condition*.

The probability distribution $\boldsymbol{\pi}$ is called an *invariant* or *stationary* distribution because it persists forever once it is established. In light of the ergodicity theorem, we may thus say the following:

- Because of the mixing property, all entries of $\boldsymbol{\pi}$ will be strictly positive.

- Starting from an arbitrary initial distribution, the state-transition probabilities of a finite-state, irreducible, and aperiodic Markov chain will converge to a stationary distribution.
- The unique stationary distribution of a finite-state, irreducible, and aperiodic Markov chain is completely independent of the initial distribution and is called steady-state distribution.

Representation of Markov Chains

The collection of all state-transition probabilities $\mathbf{Q}(i,j)$ for all transitions $(i,j) : \mathbf{A}(i,j) = 1$ satisfying the constraints $\mathbf{Q}(i,j) \geq 0$ and $\sum_{j \in \mathsf{S}} \mathbf{Q}(i,j) = 1$ for all $i \in \mathsf{S}$ determines a Markov chain. We will denote this collection of STPs by the set

$$\mathsf{Q} = \{\mathbf{Q}(i,j) : \text{for all } (i,j) : \mathbf{A}(i,j) = 1\}. \tag{2.19}$$

Note that a given set of probability measure function (pmf) Q uniquely determines the state-transition probabilities $\mathbf{Q}(i,j)$ and state probabilities $\boldsymbol{\pi}(i)$, such that Q is a complete description of a Markov chain. The set Q is a manifold.

An equivalent representation of the Markov chain is the set of all joint state-transition probabilities

$$\overline{\mathbf{Q}}(i,j) = \boldsymbol{\pi}(i)\mathbf{Q}(i,j) \quad \forall\ (i,j) : \mathbf{A}(i,j) = 1. \tag{2.20}$$

Denote this collection of joint state-transition probabilities by

$$\overline{\mathsf{Q}} = \left\{\overline{\mathbf{Q}}(i,j) : \text{for all } (i,j) : \mathbf{A}(i,j) = 1\right\}. \tag{2.21}$$

Thereby, the values $\overline{\mathbf{Q}}(i,j)$ must satisfy

$$\overline{\mathbf{Q}}(i,j) \geq 0 \quad \forall\ (i,j) : \mathbf{A}(i,j) = 1 \tag{2.22}$$

$$\sum_{(i,j):\mathbf{A}(i,j)=1} \overline{\mathbf{Q}}(i,j) = 1, \tag{2.23}$$

$$\sum_{j:\mathbf{A}(i,j)=1} \overline{\mathbf{Q}}(i,j) = \sum_{\ell:\mathbf{A}(\ell,i)=1} \overline{\mathbf{Q}}(\ell,i), \tag{2.24}$$

where (2.24) is a constraint that is equivalent to the global balance condition (2.18). A state-transition probability $\mathbf{Q}(i,j)$ is obtained from the joint state-transition probability $\overline{\mathbf{Q}}(i,j)$ in the following way:

$$\mathbf{Q}(i,j) = \frac{\overline{\mathbf{Q}}(i,j)}{\boldsymbol{\pi}(i)} = \frac{\overline{\mathbf{Q}}(i,j)}{\sum_{j' \in \mathsf{S}} \overline{\mathbf{Q}}(i,j')}. \tag{2.25}$$

We refer to Q as the set of STPs, and to $\overline{\mathsf{Q}}$ as the set of *joint* STPs. There is a one-to-one correspondence between the two descriptions Q and $\overline{\mathsf{Q}}$ of a Markov chain.

2.1.6 Output Process

Contrary to the state process $\mathcal{S}$, the output process $\mathcal{Y}$ is observable. Moreover, a realization y_k at time k is not restricted to being discrete, but can take any value in the Euclidean space, i.e. $y_k \in \mathbb{R}^{\Upsilon}$, or equivalently, $\mathsf{Y} = \mathbb{R}^{\Upsilon}$, with dimension Υ to be defined in the next subsection.

The output sequence $Y^n = (Y_1, Y_2, \ldots, Y_n)$ is a sequence of conditionally independent random variables for a given realization of the non-observable state sequence $S_0^n = (S_0, S^n) = (S_0, S_1, \ldots, S_n)$. The output Y_k is associated with the transition from a state i at time $k-1$ to a state j at time k. Its probability is conditioned on the state-transition and is by assumption time-independent. We can write

$$Pr(y_k, s_k|s_{k-1}) = Pr(y_k|s_k, s_{k-1})Pr(s_k|s_{k-1}) \quad k = 1, 2, \ldots. \tag{2.26}$$

The n-dimensional probability distribution of the output process is then given by

$$p(y^n) = \sum_{s_0^n} p(y^n, s_0^n). \tag{2.27}$$

where the summation goes over all $|\mathsf{S}|^{n+1}$ terms. Noting that

$$p(y^n, s_0^n) = p(s_0) \prod_{k=1}^{n} p(y_k, s_k|s_{k-1}) \tag{2.28}$$

and using (2.26), this yields

$$p(y^n) = \sum_{s_0^n} p(s_0) \prod_{k=1}^{n} p(y_k|s_k, s_{k-1}) p(s_k|s_{k-1}) \tag{2.29}$$

$$= \sum_{s_0^n} \boldsymbol{\pi}(s_0) \prod_{k=1}^{n} w\left(y_k|s_{k-1}, s_k; \boldsymbol{\theta}(L)\right) \mathbf{Q}\left(s_{k-1}, s_k\right) \tag{2.30}$$

where $w\left(\,\cdot\,|s_{k-1}, s_k; \boldsymbol{\theta}(L)\right)$ is the conditional observation probability distribution — parameterized by the parameter vector $\boldsymbol{\theta}(L)$ — associated with the state-transition (s_{k-1}, s_k).

If the FSM represents a communication channel, the time-k state S_k can be given by some previous channel inputs or a combination of channel inputs and internal channel states.

Example 2.6 (Linear Filter with Finite Impulse Response and AWGN)
Let

$$Y_k = \sum_{i=0}^{m} g_i X_{k-1} + Z_k$$

with fixed real coefficients g_i, with X_k taking values in $\{+1, -1\}$, and where $\mathcal{Z} = (Z_1, Z_2, \ldots)$ is white Gaussian noise. If $\mathcal{X}$ is Markov with memory m', i.e., if

$$p(x_k|x^{k-1}) = p(x_k|x_{k-m'}^{k-1}),$$

then the time-$(k-1)$ state is $S_{k-1} = (X_{k-1}, X_{k-2}, \ldots, X_{k-M})$ with total memory $M = \max\{m, m'\}$.

We observe that owing to the underlying Markov process, the output distribution of our FSM can be factored, thereby expressing the global behavior $p(y^n)$ in terms of local dependencies. This paves the way for implementing efficient trellis processing algorithms such as the sum-product algorithm.

2.1.7 Models and Probability Distributions

The purpose of a model is to represent a family of output distributions in an efficient way. This is accomplished by parameterizing the distributions. A model is therefore characterized by the distribution family and its parameters.

Throughout this thesis, we assume the conditional observation probability distribution to be a mixture of Gaussian densities. Let ξ be the *mixture degree* indicating of how many Gaussian pdfs the Gaussian mixture is composed. As a Gaussian pdf $\mathbb{G}(\cdot)$ is fully specified by its mean and variance, the Gaussian mixture pdf $p_{(i,j)}(\cdot)$ associated with the transition from state i to state j is given by the means, $\mu_{(i,j,\ell)}$, variances, $\sigma^2_{(i,j,\ell)}$, and mixture coefficients $c_{(i,j,\ell)}$, for $1 \leq \ell \leq \xi$, i.e.

$$p_{(i,j)}(\cdot) = \sum_{\ell=1}^{\xi} c_{(i,j,\ell)} \cdot \mathbb{G}(\cdot; \mu_{(i,j,\ell)}, \sigma^2_{(i,j,\ell)}) \qquad \forall \;\; \mathbf{A}(i,j) = 1. \tag{2.31}$$

The stochastic constraints

$$c_{(i,j,\ell)} \geq 0 \qquad \forall \;\; (i,j) : \mathbf{A}(i,j) = 1 \;\text{ and }\; 1 \leq \ell \leq \xi \tag{2.32}$$

$$\sum_{\ell=1}^{\xi} c_{(i,j,\ell)} = 1 \qquad \forall \;\; (i,j) : \mathbf{A}(i,j) = 1 \tag{2.33}$$

assure that the pdf is properly normalized, i.e.

$$\int_{-\infty}^{\infty} p_{(i,j)}(y)dy = 1 \qquad \forall \;\; (i,j) : \mathbf{A}(i,j) = 1. \tag{2.34}$$

The vector $\boldsymbol{\theta}(L)$ contains the means, variances, and correlation coefficients associated with the transition from state i to state j. The space of the output has dimension $\Upsilon = \big(2 \cdot \xi + (\xi - 1)\big) \cdot |\{(i,j) : \mathbf{A}(i,j) = 1\}|$. The parameter vector $\boldsymbol{\theta}(L)$ is said to be the *coordinate projection* or the realization of $\boldsymbol{\Theta}(L)$, a subspace of $\mathbb{R}^{\Upsilon}$.

The parameter vector $\boldsymbol{\theta}(L)$ is at our disposal to fit the model to particular observations. The reason for choosing a Gaussian mixture is that it belongs to the class of log-concave or elliptic symmetric densities, the most general representation of a pdf for which the Baum-Welch algorithm, the famous training procedure for finite-state HMMs, has been formulated (see [62] and references therein).

To summarize, our FSM $\mathcal{M}_{\boldsymbol{\theta}(L)}$ has order $L = |\mathsf{S}|$ and is fully determined by the set of state-transition probabilities Q and the parameter vector $\boldsymbol{\theta}(L)$.

2.1.8 Stationarity and Ergodicity

A stochastic process is an indexed sequence of random variables, i.e. $\mathcal{X} = (X_1, X_2, \ldots)$.

Definition 2.1 (Stationarity [24]) *A stochastic process $\mathcal{X}$ is said to be* stationary *if the joint distribution of any subset of the sequence of random variables is invariant with respect to shifts in the time index, i.e. if*

$$Pr(X_{i_1} = x_{i_1}, X_{i_2} = x_{i_2}, \ldots, X_{i_n} = x_{i_n}) = Pr(X_{i_1+\ell} = x_{i_1}, X_{i_2+\ell} = x_{i_2}, \ldots, X_{i_n+\ell} = x_{i_n}) \quad (2.35)$$

for every shift $\ell > 0$ (right shift) and for all $x_1, x_2, \ldots, x_n \in \mathsf{X}$.

A realization $x^n = (x_1, x_2, \ldots, x_n)$ of $X^n = (X_1, X_2, \ldots, X_n)$ is called *invariant* if $x^n = x_{1+\ell}^{n+\ell}$ for some $\ell > 0$, i.e. invariant with respect to (right) time shifts.

Definition 2.2 (Ergodic Process) *A stochastic process $\mathcal{X}$ is said to be* ergodic *if each invariant realixzation has probability either zero or one; i.e. if*

$$Pr(x^n) = 0 \text{ or } Pr(x^n) = 1 \ \forall \text{ invariant events } x^n. \quad (2.36)$$

For an engineer, ergodicity simply means that the average over time equals the average over the ensemble [38].

Theorem 2.2 (Ergodic Output) *If a finite-state Markov chain is irreducible and aperiodic (and thus stationary and ergodic), then the output process $\mathcal{Y}$ is stationary and ergodic.*

This theorem follows from the fact that the output signals are conditionally independent random variables for a given realization of the non-observable state sequence. An elegant proof can be found in [52]. We note that the HMP $\mathcal{Y}$ inherits the properties of stationarity and ergodicity from the underlying Markov process $\mathcal{S}$.

2.1.9 Sum-Product Algorithm operating on a Trellis

The sum-product algorithm subsumes many algorithms in signal processing and coding [51]. It operates on a factor graph by passing messages and is therefore also called message-passing algorithm. Here, the messages are metrics, i.e. scaled versions of a-posteriori probabilities.

The factorization of (2.30) can be expressed by the graph in Fig. 2.4. This graph is a so-called Forney-style factor graph [32, 53]. A factor graph is obtained by adding circles on each branch [51]. The input sequence x^n is not shown in (2.30) as it is usually part of the state sequence s_0^n.

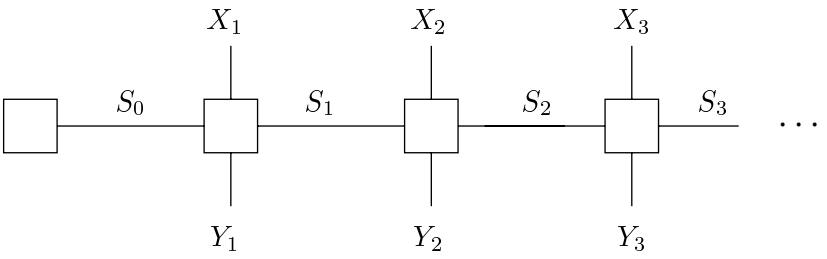

Figure 2.4: The Forney-style factor graph of (2.30).

We will assume that there is a forward "metric"

$$\mu_{\mathrm{f}}(b_k) \propto Pr(S_k = j, Y_k = y_k | S_{k-1} = i) \tag{2.37}$$

assigned to each trellis branch b_k, and that the metric of a path $b^n = (b_1, b_2, \ldots, b_n)$ is defined as $\mu_{\mathrm{f}}(b^n) \triangleq \prod_{k=1}^{n} \mu_{\mathrm{f}}(b_k)$.

Example 2.7 (Branch Metric in the Presence of AWGN)
Assume that the output of an FSM is corrupted by AWGN with variance σ^2. The time-k branch metric $\mu_{\mathrm{f}}(b_k)$ associated with the transition from state i at time $k-1$ to state j at time k becomes

$$\begin{aligned}\mu_{\mathrm{f}}(b_k) &= Pr(S_k = j, Y_k = y_k | S_{k-1} = i) \\ &= Pr(S_k = j | S_{k-1} = i) \cdot Pr(Y_k = y_k | S_k = j, S_{k-1} = i) \\ &= \mathbf{Q}(i,j) \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-(y_k - v_k)^2/(2\sigma^2)}\end{aligned}$$

where v_k is the noiseless output of the FSM associated with the state-transition (i,j) and y_k is a noisy output at time k of the FSM.

We will assume that the underlying factor graph is always a trellis (graph without loops) and that the update schedule consists of a forward (left-to-right) and a backward (right-to-left) recursion through the trellis. With this topology and this update schedule, the sum-product algorithm becomes the Bahl-Cocke-Jelinek-Raviv algorithm (BCJR algorithm [8]).

The forward recursion and the backward recursion of the sum-product algorithm are completely independent[2]. The forward recursion computes state "metrics" $\mu_{\mathrm{f}}(s_1)$, first, for all $s_1 \in \mathsf{S}_1$, then for all $s_2 \in \mathsf{S}_2$, and so on up to S_n. At time zero, there exists a state $s'_0 \in \mathsf{S}_0$ such that $\mu_{\mathrm{f}}(s'_0) = 1$ and $\mu_{\mathrm{f}}(s_0) = 0 \quad \forall\, s_0 \in \mathsf{S}_0 \backslash \{s'_0\}$. Depending on the context, other initializations may also make sense. For $k = 1, 2, \ldots, n$, $\mu_{\mathrm{f}}(s_k)$ is computed for all $s_k \in \mathsf{S}_k$ according to

$$\mu_{\mathrm{f}}(s_k) = \varphi_k \sum_{b_k : \mathrm{rst}(b_k) = s_k} \mu_{\mathrm{f}}\big(\mathrm{lst}(b_k)\big) \mu_{\mathrm{f}}(b_k). \tag{2.38}$$

In words, apart from the scaling factor φ_k, the metric of some state s_k is the sum — over all branches that end in s_k — of the product of the branch metric and the metric of the starting state of that branch. The scale factor φ_k may be chosen freely; its purpose is to prevent the state metrics from tending to zero or to infinity as k increases.

By forward recursion, the state metrics at time-k can be expressed through time-$(k-2)$ state metrics in the following way

$$\begin{aligned}\mu_{\mathrm{f}}(s_k) = \varphi_k & \sum_{b_k : \mathrm{rst}(b_k) = s_k} \mu_{\mathrm{f}}(b_k) \\ & \cdot \underbrace{\left(\varphi_{k-1} \sum_{b_{k-1} : \mathrm{rst}(b_{k-1}) = \mathrm{lst}(b_k)} \mu_{\mathrm{f}}(b_{k-1}) \mu_{\mathrm{f}}\big(\mathrm{lst}(b_{k-1})\big) \right)}_{\mu_{\mathrm{f}}\left(\mathrm{lst}(b_k)\right)}.\end{aligned} \tag{2.39}$$

[2]For a "forward-only" way to organize the computation, see [54].

The backward recursion is identical to the forward recursion except for the reversal of "time"; it computes state metrics $\mu_{\mathrm{r}}(s_{k-1})$ for every $s_{k-1} \in \mathsf{S}_{k-1}$ according to

$$\mu_{\mathrm{r}}(s_{k-1}) = \beta_k \sum_{b_k:\mathrm{lst}(b_k)=s_{k-1}} \mu_{\mathrm{r}}\big(\mathrm{rst}(b_k)\big)\mu_{\mathrm{f}}(b_k) \tag{2.40}$$

with scale factor β_k (which may be chosen freely) and with a state $s'_n \in \mathsf{S}_n$ such that $\mu_{\mathrm{r}}(s'_n) = 1$ and $\mu_{\mathrm{r}}(s_n) = 0 \quad \forall\ s_n \in \mathsf{S}_n \backslash \{s'_n\}$. These forward and backward computation steps are visualized in Fig. 2.5.

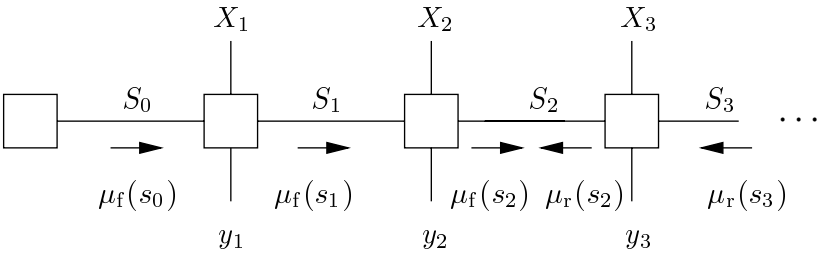

Figure 2.5: Computing forward and backward metrics in the Forney-style factor graph representing (2.30).

In principle, the forward and backward metrics can be any number associated with the state s_k and s_{k-1} respectively. For our purpose, these metrics are scaled versions of the following probabilities

$$\mu_{\mathrm{f}}(s_k) \propto Pr(S_k = s_k, Y^k = y^k) \tag{2.41}$$

$$\mu_{\mathrm{r}}(s_{k-1}) \propto Pr(Y_k^n = y_k^n | S_{k-1} = s_{k-1}) \tag{2.42}$$

From (2.39), (2.41), and (2.42), it follows that the probability of the output sequence, i.e. $p(y^n)$, can be given (upon a factor of proportionality) either by the forward recursion

$$p(y^n) \propto \sum_{s_n \in \mathsf{S}_n} \mu_{\mathrm{f}}(s_n) = \prod_{k=1}^{n} \varphi_k \sum_{\text{all paths } b^n} \mu_{\mathrm{f}}(b^n), \tag{2.43}$$

or alternatively by the backward recursion

$$p(y^n) \propto \sum_{s_0 \in \mathsf{S}_0} \mu_{\mathrm{r}}(s_0) = \prod_{k=1}^{n} \beta_k \sum_{\text{all paths } b^n} \mu_{\mathrm{f}}(b^n). \tag{2.44}$$

The main properties of the sum-product algorithm are the following. First, for every state $s'_k \in \mathsf{S}'_k$ we have the *a-posteriori state metric*

$$\mu_{\mathrm{tot}}(s'_k) \triangleq \mu_{\mathrm{f}}(s'_k) \cdot \mu_{\mathrm{r}}(s'_k) \tag{2.45}$$

$$= \prod_{i=1}^{k} \varphi_i \prod_{j=k+1}^{n} \beta_j \sum_{\text{all paths } b^n \text{ through } s'_k} \mu_{\mathrm{f}}(b^n). \tag{2.46}$$

Secondly, the *a-posteriori state-transition metric* for every branch $b'_k \in \mathsf{B}'_k$ is

$$\mu_{\text{tot}}(b'_k) \triangleq \mu_{\text{f}}\big(\text{lst}(b'_k)\big) \cdot \mu_{\text{f}}(b'_k) \cdot \mu_{\text{r}}\big(\text{rst}(b'_k)\big) \tag{2.47}$$

$$= \prod_{i=1}^{k-1} \varphi_i \prod_{j=k+1}^{n} \beta_j \sum_{\text{all paths } b^n \text{ through } b'_k} \mu_{\text{f}}(b^n). \tag{2.48}$$

The a-posteriori state metric and the a-posteriori state-transition metric are proportional to the a-posteriori state probability and the a-posteriori state-transition probability respectively, i.e.

$$\mu_{\text{tot}}(s'_k) \propto Pr(S_k = s'_k | Y^n = y^n) \tag{2.49}$$

$$\mu_{\text{tot}}(b'_k) \propto Pr(S_{k-1} = \text{lst}(b'_k), S_k = \text{rst}(b'_k) | Y^n = y^n). \tag{2.50}$$

2.2 Magnetic Recording Systems as Communication Channels

The essential elements of a magnetic recording system are the write head, the magnetic medium, and the read head. The current into the write head induces a magnetization pattern on the track directly below the write head. This pattern is arranged in equidistant cells on tracks that are concentrically aligned around the center and represent the digital data stored.

A magnetic recording system can be viewed as a communication channel. Communication stands for reliable transmission of data in space from here to there, whereas magnetic recording involves preserving information in time from now to then.

The magnetic recording process is inherently nonlinear because of the hysteresis effect. The recording channel is linearized by artificially constraining the write current to the two extreme levels such that the magnetic material always saturates. This so-called saturation recording [76] eliminates the hysteresis effect at the cost of the input being binary and thus clearly differentiates the magnetic recording channel from ordinary communication links.

2.2.1 Fundamentals of the Magnetization Process

According to Faraday's law, every electric current generates a magnetic field $\overrightarrow{H}_{\text{m}} = H_{\text{m}} \cdot \overrightarrow{e}$. If this field penetrates ferromagnetic material, the material is magnetized with a certain strength, which is denoted by the magnetization strength H_{m}. Magnetizing a magnetic medium by an applied *external* field is a complicated process because the magnetization is a nonlinear

function of the applied magnetic field, and the present magnetization depends on this field and the previous magnetization. The trajectory of the medium magnetization as a function of the magnetic field strength H_m is usually visualized by a hysteresis loop (see Fig. 2.6).

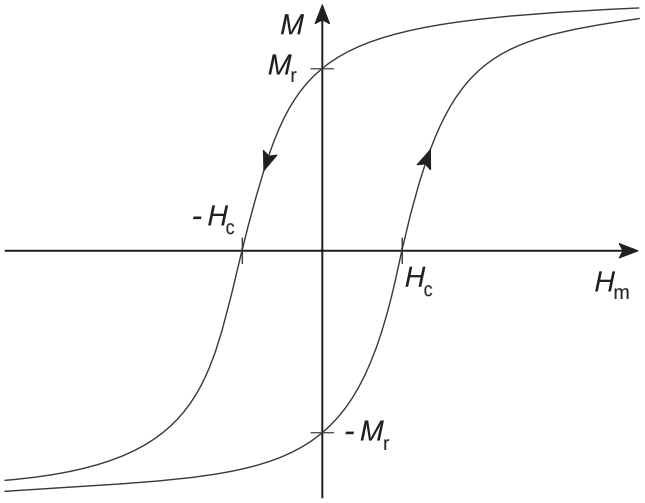

Figure 2.6: Hysteresis loop.

As the magnetization strength H_m is increased, the medium becomes magnetized in the same direction. At a certain strength of the magnetic field, the medium magnetization saturates and remains constant. This phenomenon is explained by noting that ferromagnetic material consists of magnetic domains of finite size. These domains are called *grains*. A magnetic grain, or magnetic domain, is a local region of the material which encounters a homogenous magnetization. If the magnetic orientations of the grains are completely random, the total magnetization is locally neutralized and the material is not magnetized.

By applying an external magnetic field, the number of grains magnetized in the direction parallel to the applied field increases. Because of grain interaction, this process is nonlinear. The magnetization of the medium is saturated when all grains point in the same direction. After removing the external field, not all grains switch back into their original, random direction, resulting in a certain residual *medium magnetization*. The amount of medium magnetization is termed the *remanence* or *remanent state* M_r. To change the magnetization orientation, a reverse field needs to be applied. The minimal strength of this field is referred to as the *coercivity* H_c. Coercivity and remanence are individual material constants. For magnetic disks, materials with high coercivity and high remanence are preferable. They are called *hard* magnetic materials. A high coercivity prevents incidental demagnetization by an external field (induced for instance by the room temperature). A high remanence leads to high read-back signal amplitudes. More information on the magnetization process with respect to magnetic recording can be found in [16].

2.2.2 Ideal Write Process

The current into the write head induces a magnetization pattern on the track directly below the write head. For saturation recording, the current $x(t)$ is a two-level waveform taking on values $+x_k$ and $-x_k$, resulting in an alternating (left/plus and right/minus) polarization of the underlying medium, i.e.

$$x(t) = \sum_k x_k \cdot u(t - kT) \quad \text{with} \quad u(t) = \begin{cases} 1 & \text{for} \quad t \in [0, T)\,. \\ 0 & \text{else.} \end{cases} \tag{2.51}$$

In the sequel, we assume $-x_k = -1$ and $+x_k = +1$ (see Fig. 2.7).

The magnetization changes as a function of the location ζ along the track, but does not change abruptly in response to the applied field. Thus, the transition is assigned a certain width. The widely used Williams–Comstock model for saturation recording [76] defines a *tanh*-like average magnetization profile $m_0(\zeta)$ according to

$$m_0(\zeta) = M_\mathrm{r} \tanh \frac{2\zeta}{a\pi}, \tag{2.52}$$

where a denotes the so-called transition-width parameter. For a position ζ far away from the transition, we experience the magnetization of the remanent state, i.e.

$$m_0(\zeta \to +\infty) = -m_0(\zeta \to -\infty) = M_\mathrm{r}. \tag{2.53}$$

The transition-width parameter a is a measure of the transition width. This can be seen from

$$\frac{\mathrm{d}}{\mathrm{d}\zeta} m_0(\zeta)|_{\zeta=0} = \frac{2M_\mathrm{r}}{\pi a}. \tag{2.54}$$

A useful equation for a is given in [16] by

$$a = c\sqrt{\frac{M_\mathrm{r}\delta\sqrt{d(d+\delta)}}{H_\mathrm{c}}}, \tag{2.55}$$

where c is a constant ranging from 0.6 to 0.9. The transition-width parameter a can either be decreased by reducing the medium thickness δ and the head-to-surface separation d, or by increasing the coercivity H_c of the medium.
For the time being, we assume an ideal write process and set the transition width a to zero. We discuss the nonideal write process in Chapter 4.

2.2.3 Ideal Read Process

If we assume an optimum synchronization of the disk rotation with the read channel electronics, the medium appears to be moving at a constant velocity under the read head. Thus,

the equations describing the medium magnetization $m_0(\zeta)$ and the read-back signal can be expressed either in terms of a distance ζ to an arbitrary reference point, or in terms of the time t which is required to move the medium by the distance ζ. Both are related to each other by the linear velocity ν of the disk such that $\zeta = \nu \cdot t$. Note also that we only deal with reading a single track or even a single sector, so that a one-dimensional expression of ζ or t satisfies our needs.

Although the areal density depends on both cell length and track width, we focus only on the cell length. This is justified by the general practice of "writing wide and reading narrow", which allows adjacent tracks to be neglected in a first approximation. The cell length ζ_{cell} is determined by the bit spacing parameter T, i.e. $\zeta_{\text{cell}} = \nu \cdot \big(kT - (k-1)T\big) = \nu \cdot T$.

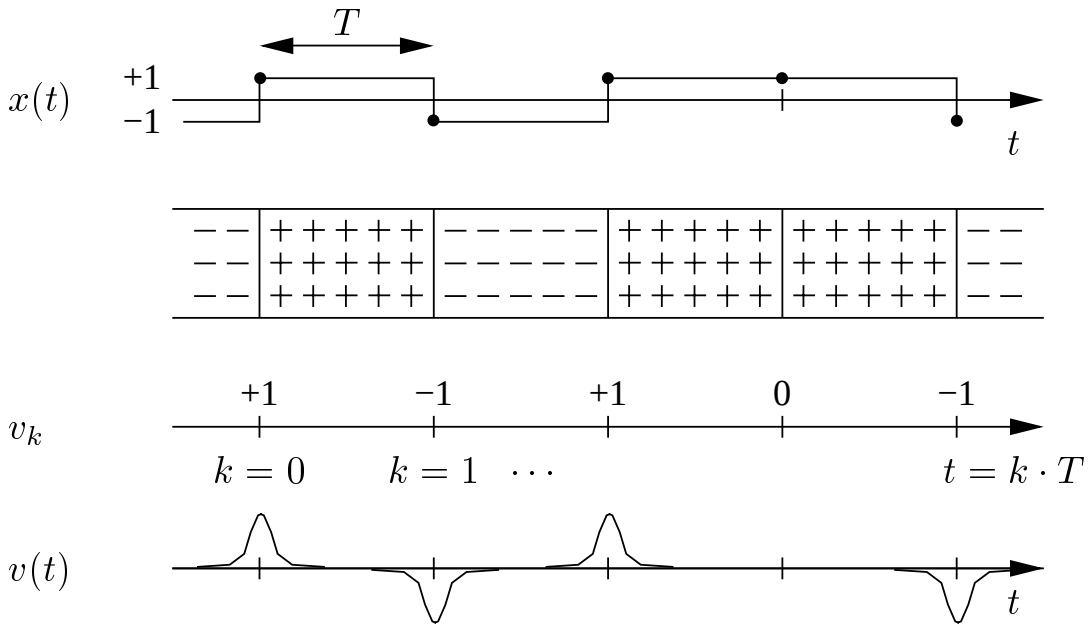

Figure 2.7: From top to bottom: Write signal $x(t)$, magnetized medium, sequence of recorded transitions $\{v_k\}$, and read-back signal $v(t)$.

During the read-back process, not the medium magnetization but rather the magnetic transitions, the "derivatives" of the medium magnetization, are sensed by the read-back head. Therefore, an isolated transition results in a certain shape $g(t)$ of the read-back signal, which is called transition response or *step response*. The specific shape varies according to the read-head types. When sensing a number of transitions, the individual transition responses will superpose linearly. Moreover, as the medium has been saturated during the write process, the read-back signal ideally is independent of the previous magnetization of the disk medium.

Formally, the recorded transition at time k is denoted by v_k, where

$$v_k = \begin{cases} 0 & \text{no transition at time } t = kT \\ \pm 1 & \text{otherwise.} \end{cases} \tag{2.56}$$

This notation corresponds directly to the sequence of magnetic transitions, and the sign of an element v_k denotes the direction of the transitions (and of the polarization). Note that $\{v_k\}$ is a correlated sequence, and related to the uncorrelated sequence $\{x_k\}$ of write current amplitudes by

$$v_k = \frac{1}{2}(x_k - x_{k-1}) \tag{2.57}$$

with initial condition $x_0 = -1$. With these assumptions, we obtain a linear model for the read-back channel. We can write the read-back signal as

$$v(t) = \sum_k v_k g(t - kT) \tag{2.58}$$

$$= \sum_k x_k h(t - kT), \tag{2.59}$$

where

$$h(t) = \frac{1}{2}\left(g(t) - g(t - T)\right). \tag{2.60}$$

We note that $h(t)$ represents the effective impulse response of the magnetic recording channel as it corresponds to the response of head and medium to a rectangular pulse, i.e. to exactly two subsequent transitions (called dibit). In the literature, $h(t)$ is commonly termed *dipulse*. The linear channel model and its signal waveforms are shown in Fig. 2.8 and Fig. 2.7 respectively.

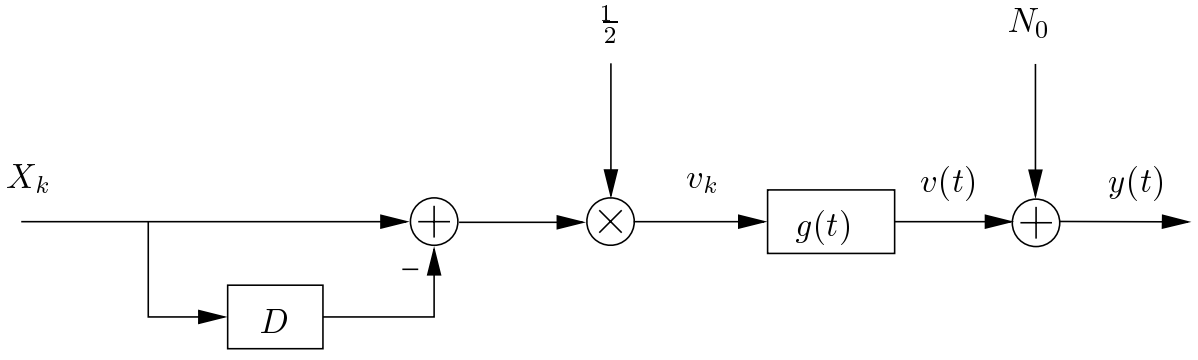

Figure 2.8: Linear channel model.

The particular shape of the transition response $g(t)$ depends on the read-head type. Basically, there are two types. Until a few years ago, inductive heads were prevalent in disk-drives.

Inductive heads operate on the principle of magnetic induction, i.e. the change of magnetic flux generates a voltage in the head coil. Nowadays, magnetoresistive (MR) read-heads and their derivative, the giant magnetoresistive (GMR) read-heads, are standard in products. Their operation is based on the change of the resistivity of certain metals in the presence of a magnetic field. This effect is called *magnetoresistance*.

As MR heads measure directly the flux from the medium, the read-back signal is fundamentally independent of the linear velocity of the rotating disk [16]. Thus, almost identical signal levels are generated at the outer and the inner track of the disk drive. Moreover, a sufficiently high output voltage is achieved with small and slowly rotating disks that operate at low linear velocity. This feature of the MR element was vital for the development of low-power notebook drives. Another significant advantage of MR read-elements is their low inductance, which leads to an increase of the read-back signal bandwidth. The most important advantage, however, is that MR heads generate a read-back signal with higher SNR than the old inductive heads do. The disadvantages of MR heads are that they produce a slightly broader pulse than inductive heads do and that the read-back pulse is asymmetric. Nevertheless, MR read-heads are considered a major breakthrough in magnetic recording towards higher areal densities.

For shielded, biased, and thus linearized MR read-heads, analytical expressions have been derived for the transition response in [16]. The shape of the read-back transition response is well approximated by the Lorentzian pulse, which is determined by a single parameter, PW50, the pulse width at 50% amplitude. The Lorentzian read-back pulse is given by

$$g(t) = \frac{1}{1 + \left(\frac{2t}{\mathrm{PW50}}\right)^2}, \tag{2.61}$$

with the pulse width in a first approximation

$$\mathrm{PW50} = 2(a + d) \tag{2.62}$$

as given in [16]. The pulse width can be narrowed by decreasing the transition-width, a, and/or the head-to-surface distance, d, of the read-back head.

The ratio PW50/T, where T is the data rate (or bit spacing parameter), is a measure of the *normalized linear density* in a hard-disk system. It is the single most important parameter to characterize the channel in a magnetic recording system. A small PW50/T causes less dispersion and therefore less ISI. It can be achieved, for a given rotation speed ν, either by high-quality magnetic materials and read-heads (i.e. small PW50) or alternatively by a low data rate (i.e. a large T).

2.2.4 Simple Channel Models

The magnetic recording channel is an inherently bad channel, as it contains ISI due to the overlapping read-back pulses, a spectral null at frequency zero (DC) due to acting as a differentiator, and strong attenuation at high frequencies due to the Lorentzian read-back waveform.

Complete removal of the ISI is therefore not possible without strong noise boosting at DC and the Nyquist frequency. Moreover, the necessary receive filter can often physically not be realized. The remedy is to control the ISI by shaping the channel to a target channel. The desired properties of the target channel are spectral nulls at DC and the Nyquist frequency, and a finite impulse response. Hence, ISI is introduced in a deterministic form that is favorable for subsequent signal processing.

For a Lorentzian pulse with a normalized linear density of $\mathrm{PW50}/T = 2.0$, little equalization is required to force the magnetic recording channel to match a class-4 partial-response (PR4) channel [50] with impulse response $g(D) = (1 - D^2)$. Moreover, this target polynomial is desirable because it enables extremely efficient sequence detection by means of the Viterbi algorithm (interleaved operation mode). The technique of shaping the channel to a desired PR polynomial with subsequent Viterbi detection has been commercialized in the early 90's under the name of partial-response maximum-likelihood detection (PRML) [23].

At higher normalized linear densities, the channel spectrum becomes quite different from the PR4 target channel. The main part of the spectrum is bent towards DC. The necessary equalization to the PR4 channel requires strong equalization schemes and leads to undesired noise enhancement at high frequencies. This noise boosting is substantial and detrimental to the performance of PRML systems. Moreover, the noise is no longer white but becomes strongly colored after equalization. By allowing the PR target polynomial to take on non-integer coefficients, a better match to the channel is possible, requiring less equalization. As this polynomial is obtained via prediction theory, the sequence detection approach is termed noise-predictive maximum-likelihood (NPML) detection [22,27]. Many of IBM's current products use NPML detection technology for reliable recovery of data from read-back signals. The target channel typically has sixteen states and can be given as

$$g(D) = (1 - D^2) \cdot (1 + p_0 D + p_1 D^2), \tag{2.63}$$

where p_0 and p_1 are the predictor coefficients.

To summarize, channel models used in today's products consist of a bipolar discrete-time input X_k, a linear finite-impulse response (FIR) channel with memory four and AWGN at the output. They give rise to aperiodic and irreducible FSMs whose output distribution is a

combination of Gaussian densities determined by means that depend on the state-transition and a single variance.

2.3 Information Theory for Memoryless and ISI Channels

We briefly summarize well-known results in information theory that will be needed in the next chapter and follow thereby the book of Cover and Thomas [24] very closely.

Entropy, Divergence, and Mutual Information

Definition 2.3 (Entropy of a Discrete Random Variable) *The entropy of a discrete random variable X with pmf $p_X(x) = p(x)$ (the subscript will be omitted) is given by*

$$H(X) \triangleq -\sum_{x \in \mathrm{supp}(p_X)} p(x) \log_{\mathrm{b}} p(x). \tag{2.64}$$

The choice of the base "b" determines the unit. When $b = 2$, the unit is called *bit* (a word suggested to Shannon by J. W. Tukey as the contraction of "binary digit"). When $b = e$, the only other base commonly used in information theory, the unit is called *nat*. Because $\log_2(e) \approx 1.443$, it follows that one nat equals about 1.443 bits. In the sequel, we will use (without loss of generality) the logarithm to the base $b = 2$ and the unit is thus a "bit".

Definition 2.4 (Conditional Entropy of two Discrete Random Variables) *The entropy of a discrete random variable X conditioned on a discrete random variable Y is given by*

$$H(X|Y) \triangleq -\sum_{x,y \in \mathrm{supp}(p_{XY})} p(x,y) \log_2 p(x|y). \tag{2.65}$$

Differential entropies and conditional differential entropies of continuous random variables are defined by replacing the summation with an integration. They are denoted by the lower-case letter "h", i.e. $h(X)$ and $h(X|Y)$, respectively.

Definition 2.5 (Discrete Memoryless Channel (DMC)) *A discrete memoryless channel consists of two finite sets X and Y and a collection of pmfs $w(y|x)$, one for each $x \in \mathsf{X}$, such that for every x and y, $w(y|x) \geq 0$, and for every x, $\sum_y w(y|x) = 1$. The pmf $w(y|x)$ is termed forward channel law, and X is the input and Y the output of the channel.*

Definition 2.6 (Mutual Information) *The mutual information between the input X and output Y of a DMC with input distribution $q(\cdot)$ and forward channel law $w(\cdot|\cdot)$ is defined as*

$$I(X;Y) \triangleq \sum_{x,y\in\mathrm{supp}(p_{X,Y})} p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)} \tag{2.66}$$

$$= \sum_{x,y\in\mathrm{supp}(q_X w_{Y|X})} q(x)w(y|x)\log_2\frac{w(y|x)}{r(y)} \tag{2.67}$$

$$= \sum_{x,y\in\mathrm{supp}(r_Y v_{X|Y})} r(y)v(x|y)\log_2\frac{v(x|y)}{q(x)} \tag{2.68}$$

where $r(y)$ is the output distribution defined as

$$r(y) \triangleq \sum_{x\in\mathrm{supp}(q_X)} q(x)w(y|x) \quad \forall y \in \mathsf{Y} \tag{2.69}$$

and $v(\cdot|\cdot)$ is the backward channel law[3] *defined as*

$$v(x|y) = \frac{w(y|x)q(x)}{r(y)} \quad \text{with } r(y) > 0, \quad \forall y \in \mathsf{Y}. \tag{2.70}$$

In terms of entropies, we can write the mutual information as

$$I(X;Y) = H(X) - H(X|Y) \tag{2.71}$$

$$= H(Y) - H(Y|X). \tag{2.72}$$

For continuous random variables differential entropies are used.

For a DMC, $H(Y)$ is a concave and $H(Y|X)$ is a linear function (i.e. concave and convex) of the input probability distribution $q(\cdot)$ [34]. Hence, the mutual information of a DMC is a concave function of the input probability distribution $q(\cdot)$.

Definition 2.7 (Divergence between Two Probability Distributions) *The divergence is a measure of the "distance" between two probability distributions $p(\cdot)$ and $q(\cdot)$. It is given by*

$$D\left(p(\cdot)||q(\cdot)\right) \triangleq \sum_{x\in\mathrm{supp}(p_X)} p(x)\log_2\frac{p(x)}{q(x)} \geq 0. \tag{2.73}$$

For continuous random variables, the condition that $q(x)$ is nonzero in $\mathrm{supp}(p_X)$, equals the condition that $\mathrm{d}\,p_X(\cdot)/\mathrm{d}\,q_X(\cdot)$, the Radon-Nikodym derivative, exists (which we will always assume). In the sequel, we will omit $\mathrm{supp}(\cdot)$ in the summation.

The divergence is always positive with equality if and only if $p(\cdot) = q(\cdot)$. It is a pseudo-distance, as it is not symmetric, i.e. in general $D\left(p(\cdot)||q(\cdot)\right) \neq D\left(q(\cdot)||p(\cdot)\right)$. The divergence is also called "relative entropy" or "Kulback-Leibler distance".

[3] Note that $v(\cdot|\cdot)$ depends on the input distribution $q(\cdot)$.

Example 2.8 (Mutual Information as Divergence)
The mutual information between the input X and output Y of a DMC can be rewritten as a divergence in the following way

$$\begin{aligned} I(X;Y) &= \sum_x q(x) \sum_y w(y|x) \log_2 \frac{w(y|x)}{r(y)} \\ &= \sum_x q(x) D\left(w(\cdot|x)||r(\cdot)\right), \end{aligned}$$

where $w(\cdot|x)$ is a probability distribution for every x, or alternatively as follows

$$\begin{aligned} I(X;Y) &= \sum_{x,y} p_{X,Y}(x,y) \log_2 \frac{p_{X,Y}(x,y)}{q(x)r(y)} \\ &= D\left(p_{X,Y}(\cdot,\cdot)||q(\cdot)r(\cdot)\right). \end{aligned}$$

Capacity

Definition 2.8 (Capacity Formula of a DMC) *The capacity of a DMC is defined as the maximum mutual information between the input X and the output Y, i.e.*

$$C \triangleq \max_{q(\cdot)} \; I(X;Y) \tag{2.74}$$

with the constraints that $q(x) \geq 0 \;\; \forall x \in \mathsf{X}$ and $\sum_x q(x) = 1$. Any $q(\cdot)$ that maximizes (2.74) is called a capacity-achieving *input distribution. Note that such a distribution may not be unique.*

In general, there exists no analytical solution for computing the capacity. For discrete-memoryless channels, the Arimoto-Blahut algorithm [1, 18] is an elegant iterative algorithm to determine a capacity-achieving input distribution and to compute capacity. For general channels, the capacity can be found numerically by non-linear optimization techniques such as gradient search or by so-called interior point methods [19]. The Karush-Kuhn-Tucker conditions are first-order necessary conditions for local optima [14] in non-linear programming problems. They give rise to a geometric interpretation that is particularly illustrative for the discrete memoryless channel. The capacity of a DMC can be expressed as follows

$$C = \max_{q(\cdot)} \sum_x q(x) D\left(w(\cdot|x)||r(\cdot)\right) \tag{2.75}$$

with the constraints that $q(x) \geq 0 \;\; \forall x \in \mathsf{X}$ and $\sum_x q(x) = 1$. Applying Lagrange multipliers one obtains the so called Karush-Kuhn-Tucker conditions, i.e.

$$\begin{cases} D\left(w(\cdot|x)||r(\cdot)\right) = C & \text{if } q(x) > 0 \\ D\left(w(\cdot|x)||r(\cdot)\right) \leq C & \text{if } q(x) = 0. \end{cases} \tag{2.76}$$

Any input distribution $q(\cdot)$ that fulfills these conditions is a capacity-achieving input distribution.

For illustrative purposes, we consider now a DMC with a three-valued input and a three-valued output. The output distribution $w(\cdot|x)$ is a point in the simplex shown in Fig 2.9. In this case, the Karush-Kuhn-Tucker conditions require that all distributions $w(\cdot|x)$ must lie on a circle with center at $r^*(\cdot) = \sum_x w(\cdot|x)q^*(x)$ where $q^*(\cdot)$ is a capacity-achieving input distribution; those $w(\cdot|x)$ that correspond to inputs x with $q^*(x) = 0$ can also lie inside the circle. The "radius" of the circle is the capacity. Thus the summation over x in (2.75) can be neglected, and capacity can be rewritten as a divergence in the following way

$$C = \min_{r(\cdot)} \max_x D\left(w(\cdot|x)||r(\cdot)\right), \tag{2.77}$$

where $r(\cdot)$ is a distribution over the output alphabet. Therefore, choosing a fixed $r(\cdot)$, one obtains the upper bound

$$C \leq \max_x D\left(w(\cdot|x)||r(\cdot)\right). \tag{2.78}$$

This upper bound appeared in a work by Topsøe [73] and was brought to our attention by Lapidoth.

According to Boyd [19], computing capacity is a geometric program. It can be formulated either as constraint maximization problem, see (2.74), or as a constraint minimization problem, see (2.77). The latter is termed *dual* expression of capacity whereas (2.74) is called *primal* expression [19].

Referring to Fig. 2.9 and (2.77), we note that it is not possible to change the min-max operation into a max-min. Hence, we first have to select a center, i.e. $r(\cdot)$, and then choose the radius in such a way that it contains all points, i.e. the max operation. As there is only one center, the output distribution induced by a capacity-achieving input distribution is unique.

To prove the uniqueness of the output distribution, the channel capacity is expressed as follows

$$C = D\left(w(\cdot|x)||r^*(\cdot)\right) \tag{2.79}$$

where $r^*(\cdot)$ the *true* output distribution, i.e. the one obtained from the channel law $w(\cdot|\cdot)$ and a capacity-achieving input distribution $q^*(\cdot)$. If $r(\cdot)$ is different from $r^*(\cdot)$, we obtain an upper bound on capacity because

$$\Delta = \max_x D\left(w(\cdot|x)||r(\cdot)\right) - C \tag{2.80}$$

$$= \max_x \left(\sum_y w(y|x) \log_2 \frac{w(y|x)}{r(y)} \right) - \sum_x q^*(x) \sum_y w(y|x) \tag{2.81}$$

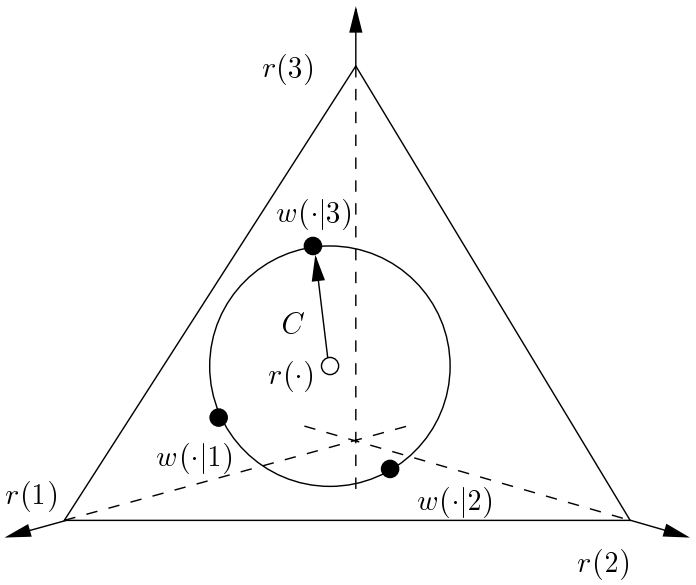

Figure 2.9: Simplex interpretation of capacity [25].

$$\geq \sum_x q^*(x) \sum_y w(y|x) \log_2 \frac{w(y|x)}{r(y)}$$

$$- \sum_x q^*(x) \sum_y w(y|x) \log_2 \frac{w(y|x)}{r^*(y)} \tag{2.82}$$

$$= \sum_x \sum_y q^*(x) w(y|x) \log_2 \frac{r^*(y)}{r(y)} \tag{2.83}$$

$$= D\left(r^*(\cdot)||r(\cdot)\right) \tag{2.84}$$

$$\geq 0 \tag{2.85}$$

with equality if and only if $r(y) = r^*(y)$ for all y.

Entropy Rate and AEP for i.i.d. Processes

Definition 2.9 (Entropy Rate) *The entropy rate of a stochastic process $\mathcal{X}$ is defined by*

$$H(\mathcal{X}) \triangleq \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n) \tag{2.86}$$

$$\triangleq \lim_{n\to\infty} H(X_n|X_{n-1}, \ldots, X_1) \tag{2.87}$$

when the limit exist. On the first line, the right-hand side expression is the per-symbol entropy rate. On the second line, the right-hand side expression is the conditional entropy rate of the last random variable given the past. For stationary stochastic processes, both are equal.

The entropy rate is the average description length for a stationary ergodic process. For stationary processes it is well defined.

Example 2.9 (i.i.d. Process)
Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed (i.i.d.) random variables. Applying the chain rule yields

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n} H(X_1, \ldots, X_n) = H(X).$$

Example 2.10 (Entropy Rate of a Markov Chain)
Because of the Markov property, the entropy rate of a Markov chain is

$$H(\mathcal{X}) = \lim_{n\to\infty} H(X_n|X_{n-1}, \ldots, X_1) = H(X_n|X_{n-1}).$$

Asymptotic Equipartition Property and Typicality

We look now at the asymptotic equipartition property (AEP) that is referred to as the law of large numbers in information theory. The AEP says that the probability assigned to a sequence of the process $\mathcal{X}$ is close to $2^{-nH(\mathcal{X})}$ with probability 1 for $n \to \infty$. This allows us to divide the set of all sequences into two sets, the set of *typical* sequences, where the sample entropy is close to the ensemble entropy, and the set of *nontypical* sequences containing all other sequences. We introduce the AEP first for i.i.d. processes, i.e. $H(\mathcal{X}) = H(X)$, and extend it afterwards to general processes (see Shannon-McMillan-Breiman theorem).

Theorem 2.3 (Asymptotic Equipartition Property (AEP) [24]) *If $X_1, X_2, \ldots, X_n$ are i.i.d. and distributed according to $p(x)$, then*

$$-\frac{1}{n} \log_2 p(X_1, X_2, \ldots, X_n) \to H(X) \tag{2.88}$$

in probability.

Proof:

$$-\frac{1}{n} \log_2 p(X_1, X_2, \ldots, X_n) = -\frac{1}{n} \sum_{i=1}^{n} \log_2 p(X_i) \tag{2.89}$$

$$= -\mathrm{E}[\log_2 p(X)] \quad \text{in} \quad \text{prob.} \tag{2.90}$$

$$= H(X). \tag{2.91}$$

■

Definition 2.10 (Typical Set [24]) *Let $X_1, X_2, \ldots, X_n$ be i.i.d.. A* typical set X_ϵ^n *with respect to the probability measure $p(x)$ is the set of sequences $(x_1, x_2, \ldots, x_n) \in \mathsf{X}^n$ having the following property:*

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, x_2, \ldots, x_n) \leq 2^{-n(H(X)-\epsilon)}. \tag{2.92}$$

As a consequence of the asymptotic equipartition property (AEP), the typical set has probability nearly 1 and all elements of the typical set are nearly equiprobable. The elements of the set are called *typical sequences* and their number is nearly $2^{nH(X)}$. All other sequences have probability nearly zero.

Shannon-McMillan-Breiman Theorem — AEP for Stationary and Ergodic Processes

The Shannon-McMillan-Breiman theorem [24] is the fundamental ergodic theorem for the sample entropy rate of stationary ergodic processes with finite alphabet, i.e. the AEP for stationary ergodic processes with finite alphabet. It states that for long sequences, the entropy rate is proportional upon a scaling factor to the logarithm of the probability of a typical sequence. Barron extended the Shannon-McMillan-Breiman theorem to densities [9] of processes with infinite alphabet. For the particular case where the process is an ergodic finite-state HMP, i.e. the output process of an FSM, Leroux provided an elegant proof in [52].

Before we focus on hidden Markov processes, we start with some general thoughts valid for any stationary and ergodic process $\mathcal{X}$. Our objective here is to gain some insight, not to provide proofs. We define the sample sequence entropy $H(x^n)$ as follows

$$H(x^n) \triangleq -\log_2 p(x^n). \tag{2.93}$$

Note that for small n the sample sequence entropy is a random variable and its value depends on the particular realization x^n. The ensemble sequence entropy, $H(X^n)$, is the average sample sequence entropy, i.e.

$$H(X^n) \triangleq \sum_{x^n \in \mathsf{X}^n} p(x^n) H(x^n) \tag{2.94}$$

$$= -\sum_{x^n \in \mathsf{X}^n} p(x^n) \log_2 p(x^n). \tag{2.95}$$

For infinitely long sequences, it converges to the entropy rate of the process $\mathcal{X}$, i.e.

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n} H(X^n). \tag{2.96}$$

For large n, almost all sequences are typical and exhibit the same probability $p(x^n)$. Thus, for large n, $H(x^n)/n$ converges to $H(X^n)/n$. But for large n, $H(X^n)/n$ converges to the entropy rate $H(\mathcal{X})$, provided that the process $\mathcal{X}$ is stationary. Therefore, we conclude that for large n the expression $H(x^n)/n$ converges to the entropy rate $H(\mathcal{X})$, i.e.

$$-\frac{1}{n} \log_2 p(x^n) \to H(\mathcal{X}) \tag{2.97}$$

with probability one provided that the process $\mathcal{X}$ is ergodic. As this holds for all typical sequences x^n, the expression $-\frac{1}{n}\log_2 p(x^n)$ is no longer a random variable but rather a constant, and one uses a capital X^n and writes

$$-\frac{1}{n}\log_2 p(X^n) \to H(\mathcal{X}) \quad \text{w. p. 1.} \tag{2.98}$$

For finite n, the averaged sample sequence entropy is thus an estimate of the entropy rate, i.e.

$$\hat{H}(\mathcal{X}) \triangleq \frac{1}{n}H(x^n) = -\frac{1}{n}\log_2 p(x^n). \tag{2.99}$$

Now, we return to a hidden Markov process $\mathcal{Y}$ with underlying state process $\mathcal{S}$.

Theorem 2.4 (Shannon-McMillan-Breiman Theorem for Hidden Markov Processes [52]) *We consider a hidden Markov process $\mathcal{Y} = (Y_1, Y_2, \ldots)$. Let the sequence Y^n be distributed according to the pdf $r(\cdot)$. Provided that the underlying finite-state Markov chain is irreducible and it holds that $E[|\log_2 p(Y_1|S_1 = j, S_0 = i; \boldsymbol{\theta}(L))|] < \infty$ for all $i, j \in \{1, \ldots, L\}$, then*

$$h(\mathcal{Y}) = -\lim_{n\to\infty} E[\log_2 r(Y_n|Y_{n-1}, Y_{n-2}, \ldots, Y_1)] \tag{2.100}$$

is finite and

$$h(\mathcal{Y}) = -\lim_{n\to\infty} \frac{1}{n}E[\log_2 r(Y^n)] \tag{2.101}$$

$$h(\mathcal{Y}) = -\lim_{n\to\infty} \frac{1}{n}\log_2 r(Y^n) \quad \text{w. p. 1.} \tag{2.102}$$

Hence, the sample sequence entropy converges with probability one to the entropy rate of the process, i.e.

$$-\frac{1}{n}\log_2 r(Y^n) \to h(\mathcal{Y}) \quad \text{w. p. 1.} \tag{2.103}$$

A proof can be found in [52]. Note: the convergence is *with* probability 1 — or equivalently "almost surely"— and not only *in* probability.

Information Rate and Capacity of Stationary and Ergodic Processes

Definition 2.11 (Information Rate) *The information rate between two stationary and ergodic processes $\mathcal{X}$ and $\mathcal{Y}$ is defined as*

$$I(\mathcal{X};\mathcal{Y}) \triangleq \lim_{n\to\infty} \frac{1}{n} I(X_1, X_2, \ldots, X_n; Y_1, Y_2, \ldots, Y_n) \tag{2.104}$$

$$= \lim_{n\to\infty} \frac{1}{n} [H(X_1, X_2, \ldots, X_n) - H(X_1, X_2, \ldots, X_n|Y_1, Y_2, \ldots, Y_n)] \tag{2.105}$$

$$= H(\mathcal{X}) - H(\mathcal{X}|\mathcal{Y}) \tag{2.106}$$

$$= h(\mathcal{Y}) - h(\mathcal{Y}|\mathcal{X}) \tag{2.107}$$

provided that the limit exists. Note: we assume here that $\mathcal{X}$ is finite-valued and $\mathcal{Y}$ continuous-valued.

Referring to $\mathcal{X}$ as input process and $\mathcal{Y}$ as output process of a communication channel, the limit in (2.104) exists if the channel law preserves the property of stationarity and ergodicity of the input process. We call such channels *ergodic channels*. All channel models used in this thesis are of this type.

Definition 2.12 (Capacity of Stationary Ergodic Channels) *The capacity between a stationary and ergodic input process $\mathcal{X}$ and a stationary and ergodic output process $\mathcal{Y}$ is defined by*

$$C \triangleq \lim_{n\to\infty} \max_{q_{X^n}} \frac{1}{n} I(X_1, X_2, \ldots, X_n; Y_1, Y_2, \ldots, Y_n). \tag{2.108}$$

2.3.1 Capacity of Memoryless Channels

Assume a DMC with discrete-time input X_k and input power 1. Assume further that a white Gaussian noise sample, Z_k, of power σ^2 is added to the noiseless output V_k at the channel output. We are interested in the maximal amount of information per symbol[4], i.e. the capacity, that can be conveyed across such a channel.

Gaussian Input Channel

Assume that the discrete input is allowed to take on any real value and the only constraint imposed on the input is the average symbol energy. In this case, capacity is given by

$$C = \frac{1}{2}\log_2\left(1 + \frac{1}{\sigma^2}\right). \tag{2.109}$$

This is the discrete-time equivalent of Shannon's famous formula via the sampling theorem [70]. The capacity-achieving input distribution is a Gaussian pdf. If logarithms to the base two are used (as we will generally do), the unit of capacity is bits/symbol.

Binary Input Channel

In addition to the average symbol constraint considered above, the channel input is now constrained to take on only the two possible symbols $+1$ and -1. By straight-forward manipulation of the capacity formula as in [61], we obtain the capacity of the binary input channel without memory and AWGN:

$$C = 1 - \frac{1}{\sqrt{2\pi}} \int_{\mathsf{Y}} e^{-y^2/2} \log_2\left(1 + \exp\left[-2\frac{y}{\sigma} - 2\frac{1}{\sigma^2}\right]\right) dy\,. \tag{2.110}$$

[4]As we only consider single-input single-output channels, the unit "information per symbol" is equivalent to the unit "information per channel use".

It is the ultimate limit for binary phase-shift keying (BSPK). The discrete-time capacity-achieving input distribution is symmetric, i.e. $Pr(X_k = +1) = Pr(X_k = -1) = 0.5$.

2.3.2 Capacity of Channels with Memory

Real communication channels are often degraded by a combination of additive, multiplicative, and input-dependent interferences of time-varying nature. Whatever the underlying physical phenomena in a particular situation are, the demodulator output signals are no longer independent events given the input sequence. The discrete-time channel is then said to have "memory". Moreover, the dependency between the output and the input is given by the channel state.

Finite-State Channels

We start by giving Gallager's definition of finite-state channels.

Definition 2.13 (Finite-State Channel [34]) *The output at time k of a finite-state channel is statistically independent of the state at time k, given the state at time $k-1$ and the input at time k, i.e.*

$$Pr(y_k, s_k|x_k, s_{k-1}) = Pr(y_k|x_k, s_{k-1})Pr(s_k|x_k, s_{k-1}). \tag{2.111}$$

If $Pr(s_k|x_k, s_{k-1}) = Pr(s_k|s_{k-1})$, i.e. if the input has no influence on the next state given the actual state, we obtain the subclass of channels with freely evolving state to which the Gilbert-Elliot channel belongs [57]. On the other hand, if $Pr(s_k|x_k, s_{k-1})$ is either 0 or 1, we have the subclass of "deterministic" channels to which discrete-time channels with finite impulse response belong.

An important subclass of FSCs is the class of *indecomposable* FSCs. For an indecomposable FSC, the effect of the starting state s_0 dies away with time.

Definition 2.14 (Indecomposable Channels [34]) *A FSC is indecomposable if for every $\epsilon > 0$, no matter how small, there exists an n' such that for all $n > n'$,*

$$|Pr(s_n|x^n, s_0) - Pr(s_n|x^n, s_0')| \leq \epsilon \tag{2.112}$$

for all s_n, x^n, s_0, and s_0'.

The probability $Pr(s_n|x^n, s_0)$ is equivalent to $Pr(s_n|s_{n-1})$ with $s_{n-1} = (x^n, s_0)$. We can think of $Pr(s_n|s_{n-1})$ as being the $(n-1, n)$-th entry of a state-transition matrix. Thus, (2.112) states that in any column all entries must be identical. Remember now that for an irreducible

and aperiodic FSM all entries of a column of $\mathbf{Q}^\infty$ are identical (see also (2.14)). Thus, the properties of being irreducible and aperiodic is equivalent to being indecomposable. We note that an indecomposable FSC can be represented by an aperiodic and irreducible FSM (mixing property).

We modify slightly the definition of the capacity of indecomposable finite-state channels from Gallager [34] for the purpose of our needs.

Definition 2.15 (Capacity Formula of Indecomposable Finite-State Channels) *Assume a finite-state channel with starting state S_0 that is driven by a discrete input X_k. The time-k discrete input X_k can take on values only in the finite set X. The capacity is given by*

$$C^{\text{FSC}} \triangleq \lim_{n\to\infty} \max_{\mathsf{K}^n} \min_{s_0} \frac{1}{n} I(X^n; Y^n | S_0 = s_0) \tag{2.113}$$

$$= \lim_{n\to\infty} \max_{\mathsf{K}^n} \max_{s_0} \frac{1}{n} I(X^n; Y^n | S_0 = s_0) \tag{2.114}$$

where $\mathsf{K}^n = \{p_{X^n}(x^n) : x^n \in \mathsf{X}^n\}$ denotes the allowed input pmfs. For bipolar input, we have $\mathsf{X}^n = \{+1, -1\}^n$.

Channels with Gaussian Input

Hirt [41] showed in his thesis that the capacity of the discrete-time Gaussian input channel with memory and *per symbol* average-energy input constraint is also the capacity of the same channel but subject to a *per block* average-energy constraint.

The optimization procedure of the input spectrum to compute capacity is preferably transformed together with the energy constraint to the frequency domain. It turns out that in the frequency domain, the optimization procedure can be done easier and has also a nice intuitive interpretation ("water-filling" technique).

Channels with Binary Input

In his seminal work [70], Shannon established many fundamental properties of *noiseless*, input-constraint communication channels. Most notably, he defined the capacity C of a discrete-time noiseless system that is characterized by the adjacency matrix $\mathbf{A}$ as

$$C = \lim_{n\to\infty} \frac{1}{n} \log_2 |\Sigma(n)| \tag{2.115}$$

where $|\Sigma(n)|$ is the maximum number[5] of sequences of length n that satisfy the constraint imposed by $\mathbf{A}$. The capacity can be computed by a combinatorial or equivalently by a prob-

[5]This implies that if there are parallel branches then these branches are countable, i.e. labeled with different symbols.

abilistic approach.

Combinatorial Approach

Assume a Markov process is given by the adjacency matrix $\mathbf{A}$, then the maximal entropy rate (the capacity) of the Markov process is given by the following theorem.

Theorem 2.5 (Capacity of a Markov Process [70]) *The maximal entropy rate of a Markov process determined by the adjacency matrix* $\mathbf{A}$ *is given by the spectral radius (largest eigenvalue) of* $\mathbf{A}$*, i.e.*

$$C = \lim_{n\to\infty} \frac{1}{n} \max_{q_{X^n}} \; H(X_1, X_2, \ldots, X_n) = \log_2 \rho(\mathbf{A}) \quad \text{bits/symbol.} \tag{2.116}$$

Proof: The entropy rate is the average growth rate of the number of sequences. The number of sequences is counted for instance by starting in all states with an all-one row vector $\boldsymbol{x}^{(0)}$. The i-th entry of the vector $\boldsymbol{x}^{(n)} = \boldsymbol{x}^{(0)} \cdot \mathbf{A}^{(n)}$ contains the number of sequences that after n time steps end in state i. Using singular value decomposition and denoting the i-th eigenvalue of $\mathbf{A}$ with λ_i, we can rewrite $\boldsymbol{x}^{(n)}$ in the following way

$$\begin{aligned}
\boldsymbol{x}^{(n)} &= \boldsymbol{x}^{(0)}\mathbf{A}^{(n)} && (2.117)\\
&= \boldsymbol{x}^{(0)}\mathbf{U}^{-1} \begin{pmatrix} \lambda_1^n & 0 & 0 & \cdots \\ 0 & \lambda_2^n & 0 & \\ 0 & 0 & \lambda_3^n & \cdots \\ \vdots & & \vdots & \ddots \end{pmatrix} \mathbf{U} && (2.118)\\
&= \boldsymbol{x}^{(0)} \cdot \mathbf{U}^{-1} \left[\sum_i \begin{pmatrix} \ddots & \vdots & & \vdots & \iddots \\ \cdots & 0 & 0 & 0 & \cdots \\ & 0 & \lambda_i^n & 0 & \\ \cdots & 0 & 0 & 0 & \cdots \\ \iddots & \vdots & & \vdots & \ddots \end{pmatrix} \right] \cdot \mathbf{U} && (2.119)\\
&= \boldsymbol{x}^{(0)} \cdot \sum_i \left[\lambda_i^n \mathbf{U}^{-1} \cdot \begin{pmatrix} \ddots & \vdots & & \vdots & \iddots \\ \cdots & 0 & 0 & 0 & \cdots \\ & 0 & 1 & 0 & \\ \cdots & 0 & 0 & 0 & \cdots \\ \iddots & \vdots & & \vdots & \ddots \end{pmatrix} \cdot \mathbf{U} \right] && (2.120)\\
&= \sum_i \lambda_i^n \tilde{\boldsymbol{x}}_i. && (2.121)
\end{aligned}$$

The vectors $\tilde{\boldsymbol{x}}_i$ do not depend on n. For n large, the number of sequences grows therefore in first order proportional to $\lambda_1 = \rho(\mathbf{A})$.

Note: as there is an singular value decomposition of $\mathbf{A}$, we can start with any $\boldsymbol{x}^{(0)}$ without affecting the result, i.e. ergodic (mixing) property of $\mathbf{A}$. ■

Example 2.11 (Capacity of the $(1,\infty)$-RLL Source)

$$C = \log_2 \rho(\mathbf{A}) = \log_2 1.6180 = 0.6942 \qquad \text{bits/symbol.}$$

Probabilistic Approach

We know that the entropy rate of a Markov process is also given by

$$H(\mathcal{X}) = H(X_n|X_{n-1}) \tag{2.122}$$
$$= \sum_{i \in \mathsf{X}_{n-1}} Pr(X_{n-1} = i) H(X_n|X_{n-1} = i). \tag{2.123}$$

Introducing states, i.e. $S_k = X_k$, yields

$$H(\mathcal{X}) = \sum_i \boldsymbol{\pi}(i) \left(- \sum_{j:\mathbf{A}(i,j)=1} \mathbf{Q}(i,j) \log_2 \mathbf{Q}(i,j) \right) \tag{2.124}$$
$$= - \sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j) \log_2 \mathbf{Q}(i,j). \tag{2.125}$$

To achieve the maximal entropy rate, we have to maximize over the set of state-transition probabilities, i.e.

$$C = \max_{\mathsf{Q}} \; H(\mathcal{X}). \tag{2.126}$$

The entropy rate of a Markov source is a concave function over the convex set of STPs Q (see Appendix C.). Shannon [70] showed that the capacity achieving state-transition probabilities are given by

$$\mathbf{Q}^*(i,j) = \frac{\boldsymbol{r}(j)}{\boldsymbol{r}(i)} \frac{\mathbf{A}(i,j)}{\rho(\mathbf{A})} \tag{2.127}$$

where $\boldsymbol{r}(i)$ is the i-th entry of $\boldsymbol{r}^T$, the right eigenvector associated with the largest eigenvalue (Perron-root) of $\mathbf{A}$.

Example 2.12 (Capacity of the $(1,\infty)$-RLL Source (cont.))
The two eigenvalues of the $(1,\infty)$-RLL source are

$$\rho_1 = 1.6180 \quad \rho_2 = -0.6180$$

with corresponding right eigenvectors

$$\boldsymbol{r}_1^T = (+0.8507 \quad +0.5257) \quad \boldsymbol{r}_2^T = (+0.5257 \quad -0.8507)\,.$$

The optimal state-transition probabilities are therefore

$$\mathbf{Q}^* = \begin{pmatrix} 0.6181 & 0.3819 \\ 1 & 0 \end{pmatrix}$$

with the corresponding state-distribution vector

$$\boldsymbol{\pi}^* = (0.7236 \quad 0.2764)\,.$$

With these numbers, the capacity of the $(1,\infty)$*-RLL source amounts to*

$$C = -\sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}^*(i)\mathbf{Q}^*(i,j)\log_2 \mathbf{Q}^*(i,j) = 0.6942 \quad \text{bits/symbol},$$

which is (of course) the same value as we have obtained with (2.116), see Example 2.11.

2.4 Problem Statement

Computing the capacity of an FSC as given by (2.113) or (2.114) is difficult because the input can be any stationary process fulfilling the constraint K^n. There is no handy description of such a process; i.e. there exists no simple parameterization, frequency interpretation, or the like.

Of course, the constraints K^n can also satisfied by a Markov source. The distinct advantage of a Markov source is that it is easily describable: a Markov source is fully specified by the set of state-transition probabilities Q. The disadvantage of using Markov sources is that the occurrence probability of a sequence generated by a Markov process cannot be arbitrary. This limitation can be circumvented by artificially increasing the Markov source over its minimal state-space realization. By doing so, the rigid structure of small Markov sources is relaxed. The flexibility is then increased to assign each input sequence its occurence probability almost "individually" (thereby approximating general stationary processes).

We assume causal channels without feedback. The channel law, i.e. the conditional distribution of the output given the input, is given as follows

$$w_{Y^n|X^n}(y^n|x_{1-m}^n) = \prod_{k=1}^{n} w_{Y_k|X_{k-m}^k}(y_k|x_{k-m}^k) = \prod_{k=1}^{n} w(y_k|x_{k-m}^k) \tag{2.128}$$

with well-defined initial condition x_{1-m}^0 for $m > 0$. As the input is finite-valued, the channel is a so-called FSC and can be represented by an FSM. Moreover, any finite-state Markov source (MS), representing the input process, and any FSC together can be represented by a single joint source/channel FSM.

We focus an indecomposable FSCs that are driven by a Markov source and call both the

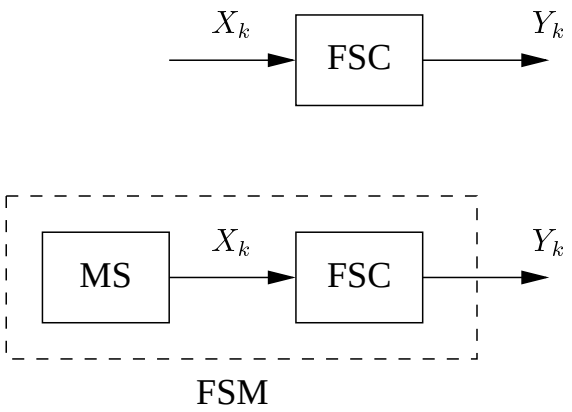

Figure 2.10: Finite-State channel (top) and FSM (bottom).

source and the channel a *finite-state model* (see Fig. 2.10). The state-transition probabilities of the input (Markov) process are free parameters that are at our disposal to maximize the information rate. Thus, we have the following definition for the capacity of FSMs.

Definition 2.16 (Capacity Formula of FSMs) *Assume that an indecomposable finite-state channel is driven by a finite-state ergodic Markov source. The capacity of the joint source/channel FSM is given in the limit as*

$$C^{\mathrm{FSM}} \triangleq \lim_{n\to\infty} \max_{\mathsf{Q}} \min_{s_0} \frac{1}{n} I(X^n; Y^n | S_0 = s_0) \tag{2.129}$$

$$= \lim_{n\to\infty} \max_{\mathsf{Q}} \max_{s_0} \frac{1}{n} I(X^n; Y^n | S_0 = s_0) \tag{2.130}$$

where Q *denotes the set of STPs imposed by the Markov source and where* s_0 *is a well-defined starting state of the joint source/channel FSM.*

Note that the capacity of an FSM is a lower bound on the capacity of the FSC, as the input of the FSC is not constrained to be a finite-state Markov process.

Thus, the objective of the thesis is to compute for a fixed Markov source at the input information rates of indecomposable FSCs, i.e. information rates of aperiodic and irreducible FSMs. In particular, we are interested in FSMs that model the magnetic recording channel.

Chapter 3

Sampling-based Computation of Information Rates

Our goal is to compute the information rate between the finite-state input process $\mathcal{X}$ (or equivalently state process $\mathcal{S}$) and the output process $\mathcal{Y}$ of an FSM. Both processes, input and output, are assumed to be stationary and ergodic. Thus, the Shannon-McMillan-Breiman theorem applies. The input process is a Markov process and satisfies the Markov property, but the output process in general does not. It is a *hidden* Markov process.

In the first two sections of this chapter, a new and practical computation method is presented for numerically evaluating estimates of the entropy rate of hidden Markov processes. It consists of sampling both input and output of the channel and of processing the resulting sequences on the trellis of the joint source/channel model. With increasing length of the sequences, the estimates converge with probability one to the limiting values predicted by the Shannon-McMillan-Breiman theorem.

This sampling-based method has been developed independently in [5], by Sharma and Singh [71], and by Pfister et al. [60]. The approach of Sharma and Singh differs from the other two approaches in that it focuses on binary-valued output processes. Thus, the theory of regenerative processes can be used rather than the Shannon-McMillan-Breiman theorem. Regenerative processes include periodic ergodic processes. However, the computation method proposed in [71] applies only to aperiodic processes and turns out to be identical to the forward recursion of the sum-product algorithm (although the authors do not seem to be aware of this). The method of Pfister et al. is the same as the one in [5].

We will show that given a finite-state Markov source, estimates of the information rate can be evaluated by the sampling-based computation method. This holds also for finite-state

Markov sources that were tailored to the channel for the purpose of maximizing the output SNR, which in turn maximizes the information rate. Such sources will deliver tight lower bounds on the capacity of the channel.

In all three papers [5,60,71], this idea was investigated and supported by numerical experiments. The latter two used gradient techniques to optimize the STPs of the Markov source and showed numerical results for a four-state input process. In [5], a systematic method was devised using the Arimoto-Blahut algorithm and exploiting the Markov property of the input process. By doing so, it was possible to optimize a 64-state input process. At the time of publication of [5], we were positively surprised that this method worked so well, but did not understand its theoretical underpinnings. We did not explain this method in [5].

Shortly afterwards, Kavčić proposed independently an elegant iterative method for maximizing the information rate [47]. Kavčić's algorithm is a highly nontrivial generalization of the Arimoto-Blahut algorithm [74] and fits nicely between Shannon's formula for the capacity-achieving state-transition probabilities of Markov sources and the Arimoto-Blahut algorithm.

Kavčić's method and our (used for the numerical results in [5]) are identical with respect to the computation of the a-posteriori state-transition weights (which will be defined shortly). Kavčić's method however differs from our method in that his update equation for the new state-transition probabilities, i.e. (3.96), contains also the eigenvector of the noisy adjacency matrix. Thus, Kavčić clearly realized the connection to Shannon's equation for the optimal state-transition probabilities [70]. We present his method in the third section of this chapter.

The fourth section of this chapter is devoted to an upper bound on the capacity of finite-state channels. Like the lower bound, this upper bound bases on computing the entropy rate of hidden Markov processes. The idea for the upper bound is from Vontobel, and numerical results show that the capacity of small channels can be well bounded [75].

In the last section of this chapter, we consider the general case where the actual channel is not a finite-state channel. In this case, the aforementioned method is applied to an auxiliary FSM that approximates the original channel. By means of the auxiliary FSM, estimates of upper and lower bounds on the information rate of the original (non finite-state) channel can be computed.

3.1 Computing Entropy Rates of Markov and Hidden Markov Processes

By exploiting the structure of the underlying Markov process, estimates of the entropy rates of Markov and hidden Markov processes are easily computed.

3.1.1 Computing the Hidden Markov Entropy Rate $h(\mathcal{Y})$

In our context, the hidden Markov entropy rate $h(\mathcal{Y})$ is the output entropy rate of our FSM. By definition, it is given for $n \to \infty$ by the expectation of $-\frac{1}{n}\log_2 r(y^n)$ with respect to the output density $r_{Y^n}(\cdot)$, i.e.

$$h(\mathcal{Y}) = \lim_{n\to\infty} \frac{1}{n} h(Y^n) \tag{3.1}$$

$$= \lim_{n\to\infty} -\frac{1}{n} \mathrm{E}_{r_{Y^n}} \left[\log_2 r(Y^n)\right]. \tag{3.2}$$

For any given block length n and any given channel output y^n, the probability $r(y^n)$ can be computed by the forward recursion of the sum-product algorithm as shown in (2.43). If the scale factor in (2.38) is chosen such that the sum of the state metrics equals one, i.e. if

$$\varphi_k = \frac{1}{\sum_{b_k \in \mathsf{B}_k} \mu_{\mathrm{f}}\left(\mathrm{lst}(b_k)\right) \mu_{\mathrm{f}}(b_k)}, \tag{3.3}$$

the state metrics are normalized at each time step. Because of this normalization, the sum of the forward state metrics at time n equals one as well. Hence, we have to divide by all normalization factors up to time n to recover $r(y^n)$, i.e.

$$r(y^n) = \frac{1}{\prod_{k=1}^{n} \varphi_k} \underbrace{\sum_{s_n \in \mathsf{S}_n} \mu_{\mathrm{f}}(s_n)}_{=1} \tag{3.4}$$

$$= \frac{1}{\prod_{k=1}^{n} \varphi_k}. \tag{3.5}$$

Thus, $r(y^n)$ equals the inverse of the product of the normalizing factors φ_k. Note: this holds also if *scaling* factors φ_k are chosen such that $\sum_{s_n \in \mathsf{S}_n} \mu_{\mathrm{f}}(s_n) = 1$.

Ensemble Expectation via the Law of Large Numbers

An estimate of $h(Y^n) = -E[\log_2 r(Y^n)]$ is obtained by the following algorithm. Take K' different channel output sequences y^n. For each such sequence, compute $r(y^n)$ as described above. Let $\phi_{k'}$ be the resulting $-\log_2 r(y^n)$ of the k'-th output sequence. The stochastic

average $\frac{1}{K'} \sum_{k=1}^{K'} \phi_{k'}$ is an estimate of $h(Y^n)$ that converges (with probability one) to $h(Y^n)$ for $K' \to \infty$ (law of large numbers).

Ensemble Expectation via Ergodicity

The above algorithm for the computation of $h(Y^n)$ was formulated mainly to gain some insight. By exploiting the ergodic property of the output process $\mathcal{Y}$, we can obtain the same estimate by a *single long sequence* y^n.

First, we note that an output sequence y^n is easily generated by feeding a random input sequence through the channel and sampling the corresponding channel output. For large n, $r(y^n)$ is computed as described above with the help of scaling factors in the forward recursion of the sum-product algorithm. Recalling (3.5), we obtain

$$r(y^n) = \frac{1}{\prod_{k=1}^{n} \varphi_k}, \tag{3.6}$$

and an estimate of the entropy rate $h(\mathcal{Y})$ is then for finite n given by

$$\hat{h}(\mathcal{Y}) = -\frac{1}{n} \log_2 r(y^n) \tag{3.7}$$

$$= \frac{1}{n} \sum_{k=1}^{n} \log_2 \varphi_k \ . \tag{3.8}$$

The right-hand side converges for $n \to \infty$ with probability one to $h(\mathcal{Y})$ because of ergodicity. An estimate of $h(\mathcal{Y})$ is thus obtained by a single long sequence y^n, and the corresponding single long forward sum-product recursion yields $r(y^n)$ as the sum of the logarithms of the scaling factors. By increasing the length n of the output sequence, the entropy rate can be estimated to any desired level of accuracy.

As the underlying FSM is by assumption irreducible and aperiodic, the influence of the starting state fades away. For long sequences n, we can therefore start in any state without affecting the estimate $\hat{h}(\mathcal{Y})$.

Algorithm 3.1 Computing the Hidden Markov Entropy Rate $h(\mathcal{Y})$

Step 0: *Fix the set of state-transitions probabilities* Q *of the joint source/channel FSM.*

Step 1: *Start in any state of the FSM and generate* y^n *for* n *large.*

Step 2: *Execute the forward sum-product algorithm on the trellis induced by the FSM with scaling factors* φ_k *as in (2.38).*

Step 3: *Compute* $\hat{h}(\mathcal{Y}) = \frac{1}{n}\sum_k \log_2 \varphi_k$.

$\hat{h}(\mathcal{Y}) \to h(\mathcal{Y})$ *for* $n \to \infty$ *with probability one.*

3.1.2 Computing the Markov Entropy Rate $H(\mathcal{X})$

The entropy rate of a Markov process is given analytically as

$$H(\mathcal{X}) = -\sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j)\log_2 \mathbf{Q}(i,j). \tag{3.9}$$

Note that (3.9) is efficient provided that $\boldsymbol{\pi}$ is known. This can be done by solving

$$\boldsymbol{\pi}\left(\mathbf{A}-\mathbf{I}\right) = \mathbf{0} \tag{3.10}$$

via Gauss-elimination.

Alternatively we can compute an estimate of $H(\mathcal{X})$ with the sum-product algorithm in essentially the same way as estimates of $h(\mathcal{Y})$ are computed. The time-k forward branch metric though changes now to $\mu_{\mathrm{f}}(b_k) = Pr\left(S_k = \mathrm{rst}(b_k) | S_{k-1} = \mathrm{lst}(b_k)\right) = \mathbf{Q}\left(\mathrm{lst}(b_k), \mathrm{rst}(b_k)\right)$.

Algorithm 3.2 Computing the Markov Entropy Rate $H(\mathcal{X})$

Step 0: *Fix the set of state transition probabilities* Q *of the Markov source.*

Step 1: *Start in any state and generate* x^n *for* n *large.*

Step 2: *Execute the forward sum-product algorithm on the source trellis with scaling factors* φ_k *as in (2.38).*

Step 3: *Compute* $\hat{H}(\mathcal{X}) = \frac{1}{n}\sum_k \log_2 \varphi_k$.

$\hat{H}(\mathcal{X}) \to H(\mathcal{X})$ *for* $n \to \infty$ *with probability one.*

The *maximal* Markov entropy rate is obtained analytically in (3.9) using the optimal state-transition probabilities

$$\mathbf{Q}^*(i,j) = \begin{cases} \frac{\boldsymbol{r}(j)}{\boldsymbol{r}(i)} \frac{1}{\rho(\mathbf{A})} & \forall (i,j) : \mathbf{A}(i,j) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

where $\boldsymbol{r}(i)$ is the i-th entry of the right Perron-vector. Plugging the optimal state-transition probabilities into (3.9) delivers the maximal entropy rate analytically as

$$H(\mathcal{X}) = \log_2 \rho(\mathbf{A}). \tag{3.12}$$

A sampling-based estimate of the maximal entropy rate is obtained with the forward recursion of the sum-product algorithm for $\mathsf{Q} = \mathsf{Q}^*$.

Computing eigenvalues and eigenvectors such as in (3.12) is for large matrices computationally very costly (in terms of computing power and storage requirements). If (as in our case) only the largest eigenvalue and corresponding eigenvector of a non-negative, primitive, real matrix $\mathbf{A}$ are of interest, the so-called *power-method* is an efficient computation method (see Appendix B). The power-method operates on the trellis spanned by the adjacency matrix $\mathbf{A}$ and computes iteratively an estimate of the largest (in magnitude) eigenvalue and corresponding eigenvector by means of the sum-product algorithm. If the trellis length goes to infinity, the estimate converges with probability one to the exact solution. Thus, computing $H(\mathcal{X})$ reveals the interwoven connection between the analytical solution and the sampling-based computation method. Computing $h(\mathcal{Y})$ according to Algorithm 3.1 can therefore be seen as a *stochastic power-method*.

3.1.3 Computing the Conditional Hidden Markov Entropy Rate $h(\mathcal{Y}|\mathcal{X})$

First, we note that

$$h(\mathcal{Y}|\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n} h(Y^n|X^n) \tag{3.13}$$

$$= \lim_{n\to\infty} \frac{1}{n} \mathrm{E}_{q_{X^n}} \left[h(Y^n|X^n = x^n)\right] \tag{3.14}$$

$$= \lim_{n\to\infty} -\frac{1}{n} \mathrm{E}_{q_{X^n}} \left[\mathrm{E}_{w_{Y^n|X^n=x^n}} \left[\log_2 w(Y^n|X^n = x^n)\right]\right]. \tag{3.15}$$

For any given sequence length n, any given channel input x^n, and channel output y^n, the probability $w(y^n|x^n)$ can be computed by the forward recursion of the sum-product algorithm. In contrast to the computation of $r(y^n)$ (see above), the sum-product algorithm now operates on a (possibly) *reduced* trellis induced by the input sequence x^n or equivalently by the corresponding state sequence s_0^n. This (possibly) reduced trellis is time-varying as x^n is a random sequence. It is given at time k by the reduced state set $\bar{\mathsf{S}}_k \subseteq \mathsf{S}_k$ and reduced branch set $\bar{\mathsf{B}}_k \subseteq \mathsf{B}_k$. The forward state and forward branch metrics at time k of the reduced trellis are denoted by $\bar{\mu}_\mathrm{f}(s_k)$ and $\bar{\mu}_\mathrm{f}(b_k)$, respectively. For $s_k \notin \bar{\mathsf{S}}_k$ and $b_k \notin \bar{\mathsf{B}}_k$, these metrics are zero. The reduced trellis is irreducible and aperiodic as well, because it originates from an irreducible and aperiodic trellis. It starts in a well-defined state determined by the initial (leftmost) state of the (complete) joint source/channel trellis. This implies that $\bar{\mu}_\mathrm{f}(s_0) = 1$ for a state $s_0 \in \bar{\mathsf{S}}_0$ and zero otherwise. The trellis is then processed from left (initial states) to right (final states), computing the forward state metrics $\bar{\mu}_\mathrm{f}(s_k)$ according to the *reduced* forward recursion rule

$$\bar{\mu}_\mathrm{f}(s_k) = \begin{cases} \sum_{b_k:\mathrm{rst}(b_k)=s_k} \bar{\mu}_\mathrm{f}\left(\mathrm{lst}(b_k)\right) \bar{\mu}_\mathrm{f}(b_k) & \text{if } s_k \in \bar{\mathsf{S}}_k \\ 0 & \text{otherwise.} \end{cases} \tag{3.16}$$

Note that because of the time-varying nature of the reduced trellis, $\bar{\mathsf{S}}_k$ depends on time. On this reduced trellis, the state metrics also tend quickly to zero, and the reduced forward recursion rule is changed to

$$\bar{\mu}_\mathrm{f}(s_k) = \begin{cases} \bar{\varphi}_k \sum_{b_k:\mathrm{rst}(b_k)=s_k} \bar{\mu}_\mathrm{f}\left(\mathrm{lst}(b_k)\right) \bar{\mu}_\mathrm{f}(b_k) & \text{if } s_k \in \bar{\mathsf{S}}_k \\ 0 & \text{otherwise,} \end{cases} \tag{3.17}$$

where $\bar{\varphi}_k$ is the time-k scale factor. If the scale factor $\bar{\varphi}_k$ is chosen such that at each time step the sum of the reduced state metrics equals one, i.e. if

$$\bar{\varphi}_k = \frac{1}{\sum_{b_k \in \mathsf{B}_k} \bar{\mu}_\mathrm{f}\left(\mathrm{lst}(b_k)\right) \bar{\mu}_\mathrm{f}(b_k)}, \tag{3.18}$$

the state metrics are normalized at each time step. To recover $w(y^n|x^n)$, $\bar{\mu}_{\mathrm{f}}(s_n)$ for $s_n \in \mathsf{S}_n$ has to be divided by the normalization factors up to time n, which yields

$$w(y^n|x^n) = \frac{1}{\prod_{k=1}^{n} \bar{\varphi}_k} \underbrace{\sum_{s_n \in \bar{\mathsf{S}}_n} \bar{\mu}_{\mathrm{f}}(s_n)}_{=1} \tag{3.19}$$

$$= \frac{1}{\prod_{k=1}^{n} \bar{\varphi}_k}. \tag{3.20}$$

Thus, $w(y^n|x^n)$ equals the inverse of the product of the normalizing factors $\bar{\varphi}_k$ on the trellis induced by the input x^n. This holds also if scaling factors are used such that $\sum_{s_n \in \bar{\mathsf{S}}_n} \bar{\mu}_{\mathrm{f}}(s_n) = 1$.

Ensemble Expectation via the Law of Large Numbers

An estimate of $h(Y^n|X^n = x^n) = -\mathrm{E}\left[\log_2 w(Y^n|X^n = x^n)\right]$ is obtained by the following algorithm. For a randomly selected input x^n, generate K' different output sequences y^n each time starting with an initial state according to the stationary state-distribution $\boldsymbol{\pi}$. For each such sequence, compute $w(y^n|x^n)$ as described above. Let $\bar{\phi}_{k'}$ be the resulting $-\log_2 w(y^n|x^n)$ of the k'-th output sequence y^n associated with the input sequence x^n. Then $\frac{1}{K'} \sum_{k'=1}^{K'} \bar{\phi}_{k'}$ is an estimate of $h(Y^n|X^n = x^n)$ that converges (with probability one) to $h(Y^n|X^n = x^n)$ for $K' \to \infty$ (law of large numbers).

An estimate of $h(Y^n|X^n)$ is obtained by averaging all $|\mathsf{X}|^n$ different input sequences x^n over $q(x^n)$, i.e.

$$h(Y^n|X^n) = \sum_{x^n \in \mathsf{X}^n} q(x^n) h(Y^n|X^n = x^n). \tag{3.21}$$

By the AEP all input sequences exhibit the same probability for n large, and thus a single long input sequence is sufficient.

Ensemble Expectation via Ergodicity

The conditional hidden Markov entropy rate can be expressed in the following way

$$h(\mathcal{Y}|\mathcal{X}) = h(\mathcal{X}, \mathcal{Y}) - H(\mathcal{X}) \tag{3.22}$$

where we note that $h(\mathcal{X}, \mathcal{Y})$[1] is a hidden Markov entropy rate as $\mathcal{Y}$ is not a Markov process. Both $h(\mathcal{X}, \mathcal{Y})$ and $H(\mathcal{X})$ can be computed by a single long sequence invoking ergodic arguments

[1] We use here a small "h" to denote the entropy rate although $\mathcal{X}$ is a finite-valued process.

as shown in the previous subsections. Thus, $h(\mathcal{Y}|\mathcal{X})$ can be computed by a single long sequence using ergodic arguments as well, i.e. it holds that

$$-\frac{1}{n}\log_2 w(Y^n|X^n) \to h(\mathcal{Y}|\mathcal{X}) \tag{3.23}$$

for $n \to \infty$ with probability one.

For large n, $w(y^n|x^n)$ is computed as described above by introducing scaling factors in the forward recursion of the sum-product algorithm. Thus, an estimate of the conditional hidden Markov entropy rate $h(\mathcal{Y}|\mathcal{X})$ is for finite n given by

$$\hat{h}(\mathcal{Y}|\mathcal{X}) = \frac{1}{n}\sum_{k=1}^{n}\log_2 \bar{\varphi}_k \ . \tag{3.24}$$

For $n \to \infty$, the right-hand side of (3.24) converges to $h(\mathcal{Y}|\mathcal{X})$ with probability one.

As the (possibly) reduced state process is by assumption indecomposable, the influence of the starting state fades away. For long sequences n, we can therefore start in any state of the (possibly) reduced trellis without affecting the estimate $\hat{h}(\mathcal{Y}|\mathcal{X})$.

Algorithm 3.3 Computing the Conditional Hidden Markov Entropy Rate $h(\mathcal{Y}|\mathcal{X})$

Step 0: *Fix the set of state-transition probabilities* Q *of the joint source/ channel FSM.*

Step 1: *Start in any state of the FSM and generate* x^n *and* y^n *for* n *large.*

Step 2: *Execute the forward sum-product algorithm on the trellis induced by* x^n *with scaling factors* $\bar{\varphi}_k$ *as in (3.17).*

Step 3: *Compute* $\hat{h}(\mathcal{Y}|\mathcal{X}) = \frac{1}{n}\sum_k \log_2 \bar{\varphi}_k$.

$\hat{h}(\mathcal{Y}|\mathcal{X}) \to h(\mathcal{Y}|\mathcal{X})$ *for* $n \to \infty$ *with probability one.*

Example 3.1 (DICODE Channel with AWGN)
Assume that the output of the DICODE channel is corrupted with AWGN. The input/state sequence induces exactly one path trough the trellis such that each $\bar{\varphi}_k$ is one sample estimate of the noise variance. The sum of the logarithms of the normalizing factors $\bar{\varphi}_k$ amounts to

$$\begin{aligned}\frac{1}{n}\sum_{k=1}^{n}\log_2\bar{\varphi}_k &= \frac{1}{n}\sum_{k=1}^{n}\log_2\left(\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y_k-v_k)^2}{2\sigma^2}}\right)^{-1}\\ &= \frac{1}{2}\log_2\left(2\pi\sigma^2\right)+\log_2\left(\exp\left[\frac{1}{2}\frac{\frac{1}{n}\sum_{k=1}^{n}(y_k-v_k)^2}{\sigma^2}\right]\right)\\ &\to \frac{1}{2}\log_2\left(2\pi e\sigma^2\right)\quad\text{w. p. 1 for } n\to\infty.\end{aligned}$$

The numerator of the last term in the penultimate line is the empirical estimate of the noise variance. It converges for large block lengths to σ^2 such that the entire expression equals the entropy of a continuous Gaussian random variable with variance σ^2.

3.1.4 Computing the Conditional Markov Entropy Rate $H(\mathcal{X}|\mathcal{Y})$

Expanding the conditional Markov entropy rate $H(\mathcal{X}|\mathcal{Y})$ in a straight-forward way yields

$$H(\mathcal{X}|\mathcal{Y}) = \lim_{n\to\infty}\frac{1}{n}H(X^n|Y^n) \tag{3.25}$$

$$= \lim_{n\to\infty}\frac{1}{n}\mathrm{E}_{r_{Y^n}}\left[H(X^n|Y^n=y^n)\right] \tag{3.26}$$

$$= \lim_{n\to\infty}-\frac{1}{n}\mathrm{E}_{r_{Y^n}}\left[\mathrm{E}_{v_{X^n|Y^n=y^n}}\left[\log_2 v(X^n|Y^n=y^n)\right]\right], \tag{3.27}$$

where the backward conditional probability $v(x^n|y^n)$ is given by

$$v(x^n|y^n) = \frac{w(y^n|x^n)q(x^n)}{\sum_{\tilde{x}^n}w(y^n|\tilde{x}^n)q(\tilde{x}^n)}. \tag{3.28}$$

For any given block length n, any given output y^n and input x^n, $v(x^n|y^n)$ can be computed by the forward *and* backward recursion of the sum-product algorithm on the joint source/channel trellis.

First, we note that the input process $\mathcal{X}$ is a Markov process. This allows us to factor the sequence x^n, (which is crucial for efficiently executing the sum-product algorithm). Because any input sequence x^n defines (by assumption) uniquely a state sequence s'^n_0 and vice versa, we have

$$v(x^n|y^n) = v(s'^n_0|y^n) \tag{3.29}$$

$$= \prod_{k=1}^{n}v(s'_k|s'_{k-1},y^n) \tag{3.30}$$

$$= \prod_{k=1}^{n}\frac{v(s'_{k-1},s'_k|y^n)}{v(s'_{k-1}|y^n)}. \tag{3.31}$$

Both, $v(s'_{k-1}, s'_k|y^n)$ and $v(s'_{k-1}|y^n)$ are computed with the forward and backward recursion of the sum-product algorithm as follows:

$$v(s'_{k-1}, s'_k|y^n) = \frac{1}{\prod_{i=1}^{k-1} \varphi_i \prod_{j=k+1}^{n} \beta_j} \mu_{\text{tot}}(b'_k), \tag{3.32}$$

provided it holds that

$$b'_k : \left(S'_{k-1} = s'_{k-1} = \text{lst}(b'_k)\right) \cap \left(S'_k = s'_k = \text{rst}(b'_k)\right), \tag{3.33}$$

and

$$v(s'_{k-1}|y^n) = \frac{1}{\prod_{i=1}^{k-1} \varphi_i \prod_{j=k}^{n} \beta_j} \mu_{\text{tot}}(s'_{k-1}), \tag{3.34}$$

with $\mu_{\text{tot}}(b'_k)$ and $\mu_{\text{tot}}(s'_{k-1})$ determined by (2.48) and (2.46), respectively; thus

$$v(x^n|y^n) = \prod_{k=1}^{n} \frac{\sum_{\text{all } b^n \text{ through } b'_k} \mu_{\text{f}}(b^n)}{\sum_{\text{all } b^n \text{ through } s'_{k-1}} \mu_{\text{f}}(b^n)}, \tag{3.35}$$

where b'_k and s'_k are induced by the input sequence x^n.

Ensemble Expectation via the Law of Large Numbers

An estimate of $H(X^n|Y^n = y^n)$ is obtained similarly as for $h(Y^n|X^n = x^n)$. As it is difficult to select y^n at random, the output sequence is first generated. For a given output sequence y^n, all possible input sequences x^n are listed in the trellis.

Thus, for a randomly selected x^n, generate K' different output sequences, each time starting in an initial state according to the stationary state-distribution. For each such output sequence compute $v(x^n|y^n)$ with the forward-backward sum-product algorithm for all input sequences as described above. With $\phi_{k''}$ being the logarithm of the a-posteriori probability of an input sequence x^n, i.e. $-\log_2 v(x^n|y^n)$, we obtain for each output sequence

$$H(X^n|Y^n = y^n) = \frac{1}{|\mathsf{X}|^n} \sum_{k''=1}^{|\mathsf{X}|^n} \phi_{k''}, \tag{3.36}$$

where the summation goes over all input sequences.

An estimate of $H(X^n|Y^n)$ is obtained from approximating the expectation over Y^n (integration) by averaging over all K' output sequences y^n collected in the set K', i.e.

$$H(X^n|Y^n) \approx \sum_{y^n \in \mathsf{K}'} r(y^n) H(X^n|Y^n = y^n) \tag{3.37}$$

with equality for $K' \to \infty$ with probability one.

Ensemble Expectation via Ergodicity

The conditional Markov entropy rate can be expressed in the following way

$$H(\mathcal{X}|\mathcal{Y}) = h(\mathcal{X},\mathcal{Y}) - h(\mathcal{Y}) \tag{3.38}$$

where we note that $h(\mathcal{X},\mathcal{Y})$ is a hidden Markov entropy rate as $\mathcal{Y}$ is not a Markov process. Both $h(\mathcal{X},\mathcal{Y})$ and $h(\mathcal{Y})$ can be computed by a single long sequence invoking ergodic arguments as shown above. Thus, $H(\mathcal{X}|\mathcal{Y})$ can be computed by a single long sequence using ergodic arguments and it holds that

$$-\frac{1}{n}\log_2 v(X^n|Y^n) \to H(\mathcal{X}|\mathcal{Y}) \tag{3.39}$$

for $n \to \infty$ with probability one.

An estimate of $H(\mathcal{X}|\mathcal{Y})$ is thus obtained by a single long output sequence y^n and the corresponding forward and backward recursion of the sum-product algorithm. As the underlying FSM is by assumption irreducible and aperiodic, the influence of the starting state fades away. For long sequences n, we can therefore start in any state without affecting the estimate $\hat{H}(\mathcal{X}|\mathcal{Y})$.

Algorithm 3.4 Computing the Conditional Markov Entropy Rate $H(\mathcal{X}|\mathcal{Y})$ — Version A

Step 0: *Fix the set of state-transition probabilities* Q *of the joint source/ channel FSM.*

Step 1: *Start in any state and generate* x^n *and* y^n*.*

Step 2: *Execute the forward and the backward recursion of the sum-product algorithm and compute the a-posteriori state-transition probabilities and a-posteriori state probabilities.*

Step 3: *Compute*

$$\hat{H}(\mathcal{X}|\mathcal{Y}) = -\frac{1}{n}\log_2 \prod_{k=1}^{n} \frac{\sum_{\text{all } b^n \text{ through } b'_k} \mu_{\mathrm{f}}(b^n)}{\sum_{\text{all } b^n \text{ through } s'_{k-1}} \mu_{\mathrm{f}}(b^n)} \tag{3.40}$$

$$= -\frac{1}{n}\sum_{k=1}^{n} \log_2 \frac{\sum_{\text{all } b^n \text{ through } b'_k} \mu_{\mathrm{f}}(b^n)}{\sum_{\text{all } b^n \text{ through } s'_{k-1}} \mu_{\mathrm{f}}(b^n)}. \tag{3.41}$$

where b'_k *and* s'_{k-1} *are induced by the input* x^n*.*

$\hat{H}(\mathcal{X}|\mathcal{Y}) \to H(\mathcal{X}|\mathcal{Y})$ *for* $n \to \infty$ *with probability one.*

Note that the computation of (3.41) is numerically problematic if the input sequence travels through some branches considerably less frequently than through others. The following computation method is mathematically equivalent, more intuitive, and numerically advantageous.

Exploiting the Markov Structure

Exploiting the Markov structure right from the beginning, we can expand the conditional Markov entropy rate as in [47]:

$$H(\mathcal{X}|\mathcal{Y}) = \lim_{n\to\infty} \frac{1}{n} H(X^n|Y^n) \tag{3.42}$$

$$= \lim_{n\to\infty} \frac{1}{n} H(S_0^n|Y^n) \tag{3.43}$$

$$= \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} H(S_k|S_{k-1}, Y^n) \tag{3.44}$$

where we assumed that X^n implies S_0^n and vice versa. A single entropy term is further expanded as follows

$$H(S_k|S_{k-1}, Y^n) = -\mathrm{E}_{p_{S_k,S_{k-1},Y^n}} \left[\log_2 Pr(S_k|S_{k-1}, Y^n)\right] \tag{3.45}$$

$$= -\sum_{(i,j):\mathbf{A}(i,j)=1} \Bigg(\boldsymbol{\pi}(i)\mathbf{Q}(i,j) \mathrm{E}_{p_{Y^n|S_k=j,S_{k-1}=i}} \left[\log_2 Pr(S_k = j|S_{k-1} = i, Y^n)\right] \Bigg). \tag{3.46}$$

In the absence of noise, Y^n determines S_0^n and thus $H(\mathcal{X}|\mathcal{Y}) = 0$. In the presence of noise, we need to compute a-posteriori probabilities. To do so, we first define

$$v_k(i,j|Y^n) \triangleq Pr(S_{k-1} = i, S_k = j|Y^n) \tag{3.47}$$

$$v_{k-1}(i|Y^n) \triangleq Pr(S_{k-1} = i|Y^n) \tag{3.48}$$

and rearrange the expectation from above as follows:

$$H(S_k|S_{k-1}, Y^n) = -\sum_{(i,j):\mathbf{A}(i,j)=1} \left(\boldsymbol{\pi}(i)\mathbf{Q}(i,j) \mathrm{E}_{p_{Y^n|S_k=j,S_{k-1}=i}} \left[\log_2 \frac{v_k(i,j|Y^n)}{v_{k-1}(i|Y^n)}\right] \right) \tag{3.49}$$

such that

$$H(\mathcal{X}|\mathcal{Y}) = -\sum_{(i,j):\mathbf{A}(i,j)=1} \left(\boldsymbol{\pi}(i)\mathbf{Q}(i,j) \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \mathrm{E}_{p_{Y^n|S_k=j,S_{k-1}=i}} \left[\log_2 \frac{v_k(i,j|Y^n)}{v_{k-1}(i|Y^n)}\right] \right). \tag{3.50}$$

To simplify this expression, we define a-posteriori state-transition weights.

Definition 3.1 (A-posteriori State-Transition Weight) *For all transitions* $(i,j) : \mathbf{A}(i,j) = 1$*, we define the a-posteriori state-transition weight in the following way:*

$$\mathbf{T}(i,j) \triangleq \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \mathrm{E}_{p_{Y^n|S_k=j,S_{k-1}=i}} \left[\log_2 \frac{v_k(i,j|Y^n)}{v_{k-1}(i|Y^n)} \right]. \tag{3.51}$$

For all transitions $(i,j) : \mathbf{A}(i,j) = 0$*, we define* $\mathbf{T}(i,j) \triangleq -\infty$*. Note: the a-posteriori state-transition weight depends on a particular source, i.e. on the set of (a-priori) STPs* Q*. We call this version of the a-posteriori state-transition weights the* conditioned version*, as the expectation is conditioned on a particular state-transition. The a-posteriori state-transition weights are collected in the matrix* $\mathbf{T}$*.*

We return now to (3.46) and expand $H(S_k|S_{k-1}, Y^n)$ in a different way, i.e.

$$H(S_k|S_{k-1}, Y^n) = -\sum_{(i,j):\mathbf{A}(i,j)=1} \left(\boldsymbol{\pi}(i)\mathbf{Q}(i,j) \mathrm{E}_{\frac{v_k(i,j|Y^n)}{p_{S_k=j,S_{k-1}=i}} r_{Y^n}} \left[\log_2 \frac{v_k(i,j|Y^n)}{v_{k-1}(i|Y^n)} \right] \right) \tag{3.52}$$

$$= -\sum_{(i,j):\mathbf{A}(i,j)=1} \left(\boldsymbol{\pi}(i)\mathbf{Q}(i,j) \mathrm{E}_{r_{Y^n}} \left[\log_2 \frac{v_k(i,j|Y^n)^{\frac{v_k(i,j|Y^n)}{\boldsymbol{\pi}(i)\mathbf{Q}(i,j)}}}{v_{k-1}(i|Y^n)^{\frac{v_{k-1}(i|Y^n)}{\boldsymbol{\pi}(i)}}} \right] \right). \tag{3.53}$$

Similarly to (3.50), this yields

$$H(\mathcal{X}|\mathcal{Y}) = -\sum_{(i,j):\mathbf{A}(i,j)=1} \left(\boldsymbol{\pi}(i)\mathbf{Q}(i,j) \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \mathrm{E}_{r_{Y^n}} \left[\log_2 \frac{v_k(i,j|Y^n)^{\frac{v_k(i,j|Y^n)}{\boldsymbol{\pi}(i)\mathbf{Q}(i,j)}}}{v_{k-1}(i|Y^n)^{\frac{v_{k-1}(i|Y^n)}{\boldsymbol{\pi}(i)}}} \right] \right). \tag{3.54}$$

We recognize the expectation term in (3.54) as the a-posteriori state-transition weight $\mathbf{T}(i,j)$ defined above. This time, the expectation is slightly different. It is *unconditioned*. Thus, we define the *unconditioned* version of the a-posteriori state-transition weight.

Definition 3.2 (Unconditioned Version of the A-posteriori State-Transition Weight) *For all transitions* $(i,j) : \mathbf{A}(i,j) = 1$*, we define the* unconditioned version *of the a-posteriori state-transition weight in the following way*

$$\mathbf{T}(i,j) \triangleq \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \mathrm{E}_{r_{Y^n}} \left[\log_2 \frac{v_k(i,j|Y^n)^{\frac{v_k(i,j|Y^n)}{\boldsymbol{\pi}(i)\mathbf{Q}(i,j)}}}{v_{k-1}(i|Y^n)^{\frac{v_{k-1}(i|Y^n)}{\boldsymbol{\pi}(i)}}} \right]. \tag{3.55}$$

For all transitions $(i,j) : \mathbf{A}(i,j) = 0$*, we define* $\mathbf{T}(i,j) \triangleq -\infty$*. The unconditioned a-posteriori state-transition weights are collected in the matrix* $\mathbf{T}$*. They are a function of the set of a-priori STPs* Q*.*

As (3.55) is mathematically equivalent to (3.51), we use the same symbol for both the conditioned and unconditioned version. The difference between the conditioned and unconditioned version of the a-posteriori state-transition weights becomes visible for n finite (see below).

In the absence of noise, the a-posteriori state-transition weight $\mathbf{T}(i,j)$ is zero as both $v_k(i,j|Y^n)$ and $v_{k-1}(i|Y^n)$ are either 0 or 1. In the presence of noise, the output sequence Y^n can in the worst case not provide any information at all about the state-transitions; meaning that $v_k(i,j|Y^n)$ and $v_{k-1}(i|Y^n)$ are both independent of Y^n. Thus, we obtain the following lemma.

Lemma 3.1 (Range of A-posteriori State-Transition Weight)

$$\log_2 \mathbf{Q}(i,j) \leq \mathbf{T}(i,j) \leq 0 \qquad \forall\ (i,j) : \mathbf{A}(i,j) = 1\ . \tag{3.56}$$

Proof: The upper limit is clear. The lower limit follows from (3.46) with $Pr(S_k = j|S_{k-1} = i, Y^n = y^n) = Pr(S_k = j|S_{k-1} = i)$. ■

The conditional Markov entropy rate is now more compactly expressed as

$$H(\mathcal{X}|\mathcal{Y}) = -\sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(j)\mathbf{Q}(i,j)\mathbf{T}(i,j). \tag{3.57}$$

Applying ergodicity, we replace the ensemble expectation in (3.55) by a single long sequence y^n and obtain the empirical counterpart as

$$\hat{\mathbf{T}}(i,j) = \frac{1}{n}\sum_{k=1}^{n} \log_2 \frac{v_k(i,j|y^n)^{\frac{v_k(i,j|y^n)}{\boldsymbol{\pi}(i)\mathbf{Q}(i,j)}}}{v_{k-1}(i|y^n)^{\frac{v_{k-1}(i|y^n)}{\boldsymbol{\pi}(i)}}}. \tag{3.58}$$

Equivalently, we can compute by means of ergodicity the *conditioned* version for the a-posteriori state-transition weight from (3.51) as follows

$$\hat{\mathbf{T}}(i,j) = \frac{1}{n(i,j)} \sum_{\substack{k=1 \\ (S_{k-1}=i,S_k=j)\in x^n}}^{n} \log_2 \frac{v_k(i,j|y^n)}{v_{k-1}(i|y^n)}. \tag{3.59}$$

where $n(i,j)$ is the number of state-transitions (i,j) that were induced by the input sequence x^n. We recognize here the same potential for numerical problems as in (3.41); because for finite n the input sequence x^n may have visited some state-transitions considerably less frequently than others.

Finally, an estimate of the conditional Markov entropy rate is expressed as follows

$$\hat{H}(\mathcal{X}|\mathcal{Y}) = -\sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(j)\mathbf{Q}(i,j)\hat{\mathbf{T}}(i,j)\ , \tag{3.60}$$

and it holds that $\hat{H}(\mathcal{X}|\mathcal{Y}) \to H(\mathcal{X}|\mathcal{Y})$ for $n \to \infty$ with probability one, because $\hat{\mathbf{T}}(i,j) \to \mathbf{T}(i,j)$ for $n \to \infty$ with probability one. Thus, we have the following algorithm:

Algorithm 3.5 Computing the Conditional Markov Entropy Rate $H(\mathcal{X}|\mathcal{Y})$ — Version B

Step 0: *Fix the set of state-transition probabilities Q of the joint source/ channel FSM.*

Step 1: *Start in any state of the FSM and generate x^n and y^n for n large.*

Step 2: *Execute the forward and the backward recursion of the sum-product algorithm and compute the a-posteriori state-transition probabilities and a-posteriori state probabilities according to (2.46) and (2.48)*

Step 3: *Compute $\hat{\mathbf{T}}(i,j)$ according (3.58) or (3.59).*

Step 4: *Compute $\hat{H}(\mathcal{X}|\mathcal{Y})$ according to (3.60).*

$\hat{H}(\mathcal{X}|\mathcal{Y}) \to H(\mathcal{X}|\mathcal{Y})$ for $n \to \infty$ with probability one.

3.1.5 Backward Sum-Product Recursion

Instead of performing a single long forward recursion of the sum-product algorithm, estimates of the entropy rates can also be computed by a single long *backward* recursion. This follows from (2.44). An estimate of the hidden Markov entropy rate $h(\mathcal{Y})$ is then obtained by

$$\hat{h}(\mathcal{Y}) = \frac{1}{n} \sum_{k=1}^{n} \log_2 \beta_k \ . \tag{3.61}$$

The same holds for computing estimates of $h(\mathcal{Y}|\mathcal{X})$ and $H(\mathcal{X})$. The conditional Markov entropy rate $H(\mathcal{X}|\mathcal{Y})$ requires forward-backward computation anyway, unless it is computed as $h(\mathcal{X},\mathcal{Y}) - h(\mathcal{Y})$.

3.1.6 Reduced-State Version

Let S'_k be a subset of the time-k states for $0 \le k \le n$. If the sum in the recursion rule in (2.38) for states $s_k \in \mathsf{S}'_k$ is modified to

$$\mu_{\mathrm{f}}(s_k) = \tilde{\varphi}_k \sum_{b_k : \mathrm{rst}(b_k) = s_k \in \mathsf{S}'_k} \mu_{\mathrm{f}}\big(\mathrm{lst}(b_k)\big) \mu_{\mathrm{f}}(b_k), \tag{3.62}$$

the sum of the final state metrics will be a lower bound on $r(y^n)$ and the corresponding estimate $\hat{h}(\mathcal{Y}) = \frac{1}{n}\sum_k \log_2 \tilde{\varphi}_k$ will be an upper bound on $h(\mathcal{Y})$ for $n \to \infty$. We have proved:

Theorem 3.2 (Reduced-State Upper Bound on $h(\mathcal{Y})$) *Omitting states from the computation in (2.38) yields an upper bound on $h(\mathcal{Y})$.*

The set S'_k may be chosen arbitrarily. An obvious strategy is to keep only a fixed number of states with the largest metrics (see also Chapter 4). By a similar argument, one may obtain a lower bound.

Theorem 3.3 (Reduced-State Lower Bound on $h(\mathcal{Y})$) *Merging states in the computation (2.38) yields a lower bound on $h(\mathcal{Y})$.*

The idea of the merging procedure is to increase the number of allowed paths in the trellis while at the same time reduce the number of states. There are various merging procedures. We will detail one in Chapter 4.

Similarly, these reduced-state bounds deliver upper and lower bounds on $h(\mathcal{Y}|\mathcal{X})$, $H(\mathcal{X})$, and $H(\mathcal{X}|\mathcal{Y})$ for $n \to \infty$.

3.2 Computing Information Rates

The algorithms presented in the first section of this chapter enable us to compute estimates of information rates of FSMs. We distinguish between a "forward-only" and a "forward-backward" method. While the former is computationally preferable, the latter is used for maximizing the information rate (see next section).

3.2.1 Forward-Only Method

The information rate between the input (or equivalently the state) process $\mathcal{X}$ and the output process $\mathcal{Y}$ of an FSM is expressed as follows

$$I(\mathcal{X};\mathcal{Y}) = h(\mathcal{Y}) - h(\mathcal{Y}|\mathcal{X}) \tag{3.63}$$

$$= \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \log_2 \varphi_k - \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \log_2 \bar{\varphi}_k \tag{3.64}$$

$$= \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \log_2 \frac{\varphi_k}{\bar{\varphi}_k} \tag{3.65}$$

where φ_k the scaling factor given by (2.38) and where $\bar{\varphi}_k$ is the scaling factor from (3.17). Thus, an estimate of the information rate is obtained for finite sequence length n by a single

long output sequence y^n followed by the computation of the scaling factors φ_k and $\bar{\varphi}_k$, i.e.

$$\hat{I}(\mathcal{X};\mathcal{Y}) = \frac{1}{n}\sum_{k=1}^{n} \log_2 \frac{\varphi_k}{\bar{\varphi}_k}. \tag{3.66}$$

Both scaling factors can be computed simultaneously with a single forward recursion of the sum-product algorithm.

3.2.2 Forward-Backward Method

The information rate is given in terms of Markov entropy rates as follows

$$I(\mathcal{X};\mathcal{Y}) = H(\mathcal{X}) - H(\mathcal{X}|\mathcal{Y}) \; . \tag{3.67}$$

Estimates of $H(\mathcal{X})$ and $H(\mathcal{X}|\mathcal{Y})$ can be computed as described in the previous section. Alternatively, the Markov property of the state process yields directly

$$I(\mathcal{X};\mathcal{Y}) = -\sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j)\Big[\log_2 \mathbf{Q}(i,j) - \mathbf{T}(i,j)\Big]. \tag{3.68}$$

An estimate of the information rate is therefore obtained by computing the empirical counterpart of the a-posteriori state-transition weights according to (3.58) or (3.59), i.e.

$$\hat{I}(\mathcal{X};\mathcal{Y}) = -\sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j)\Big[\log_2 \mathbf{Q}(i,j) - \hat{\mathbf{T}}(i,j)\Big]. \tag{3.69}$$

Noisy Adjacency Matrix

In the sequel, it will be advantageous to arrange the a-posteriori state-transition weights in a slightly different way. We start by defining the *noisy adjacency matrix* as follows:

Definition 3.3 (Noisy Adjacency Matrix) *The noisy adjacency matrix $\tilde{\mathbf{A}}$ is given by its entries $\tilde{\mathbf{A}}(i,j)$ that are defined as follows*

$$\tilde{\mathbf{A}}(i,j) \triangleq \begin{cases} 2^{\mathbf{T}(i,j)} & \forall (i,j) : \mathbf{A}(i,j) = 1 \\ 0 & \textit{otherwise} \end{cases} \tag{3.70}$$

with $\mathbf{T}(i,j)$ being the a-posteriori state-transition weight. Note that the entries of the noisy adjacency matrix, $\tilde{\mathbf{A}}(i,j)$, depend on Q.

From Lemma 3.1 follows immediately

Lemma 3.4 (Range of $\tilde{\mathbf{A}}(i,j)$)

$$\mathbf{Q}(i,j) \leq \tilde{\mathbf{A}}(i,j) \leq 1 \qquad \forall\ (i,j) : \mathbf{A}(i,j) = 1 \; . \tag{3.71}$$

By means of (3.70), the information rate of an FSM for a fixed Markov input process determined by Q can be written as

$$I(\mathcal{X};\mathcal{Y}) = \sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j)\log_2 \frac{\tilde{\mathbf{A}}(i,j)}{\mathbf{Q}(i,j)}. \tag{3.72}$$

Each valid transition (i,j) is assigned a weight $\tilde{\mathbf{A}}(i,j)$ that depends on the associated output SNR. As the state and state-transition probabilities are all smaller than one, the entries of the noisy adjacency matrix $\tilde{\mathbf{A}}$ are reduced by the noise. In absence of noise, clearly $\tilde{\mathbf{A}}(i,j) = \mathbf{A}(i,j)$.

Example 3.2 (i.u.d. Binary Input in the very low SNR Regime) *Assume that the input is i.u.d. and that the input x^n induces a unique branch sequence in the FSM. If the noise is completely dominating (very low SNR regime), then $\tilde{\mathbf{A}}(i,j) = \mathbf{Q}(i,j)$ and*

$$I(\mathcal{X};\mathcal{Y}) = \sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j)\log_2 \frac{\mathbf{Q}(i,j)}{\mathbf{Q}(i,j)} = 0. \tag{3.73}$$

For finite n, an estimate of the information rate is given by the empirical counterpart of (3.72) as follows

$$\hat{I}(\mathcal{X};\mathcal{Y}) = \sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j)\log_2 \frac{\hat{\tilde{\mathbf{A}}}(i,j)}{\mathbf{Q}(i,j)}, \tag{3.74}$$

where $\hat{\tilde{\mathbf{A}}}$ is obtained from (3.70) using $\hat{\mathbf{T}}(i,j)$s, the empirical counterparts of the $\mathbf{T}(i,j)$s.

3.2.3 Convergence Behavior

The central limit theorem (CLT) for ergodic Markov processes states that estimates of the entropy rate converge in distribution to a Gaussian random variable, with the ensemble entropy rate as mean and a variance that decays with $O(n^{-\frac{1}{2}})$ to zero [21]. More formally, let the ergodic Markov chain have the stationary state-distribution $\boldsymbol{\pi}$, then the partial sum

$$PS(n) = \sum_{k=1}^{n} f(S_k), \tag{3.75}$$

where $f(\cdot) : \mathsf{S} \to \mathbb{R}$ is any measurable function from the set of states S to the number of reals, converges in distribution to a Gaussian random variable, i.e.

$$PS(n)/\sqrt{n} \to \mathbb{G}(0,\sigma_\infty^2). \tag{3.76}$$

The function $f(\cdot)$ can be for instance the difference between the entropy rate and the time-k sample estimate, i.e.

$$f(S_k) = \left(h(\mathcal{Y}) - \log_2 \phi_k\right). \tag{3.77}$$

The variance of the partial sum of (3.77) depends on the structure, i.e. the number of states, of the underlying FSM and the conditional observation function (must be square integrable), and it is difficult to quantify.

To give some intuition, we consider the forward-backward method from above. The accuracy of the estimated information rate depends on the accuracy of the $\hat{\mathbf{T}}(i,j)$s. Assuming AWGN, i.u.d. input, and aiming at an accuracy of 1% for the estimated information rate, each branch should be visited at least 10000-times (the confidence interval decreases with $1/\sqrt{n}$). Thus, the length of the input sequence should be roughly 10000 times the number of branches of the joint/source channel model.

It was proven by Chen [21] that if the normalized sum $PS(n)/\sqrt{n}$ is bounded (as in our case), the CLT implies that the Gaussian distribution is the only possible distribution. From the Shannon-McMillan-Breiman theorem, we know that for stationary and ergodic Markov processes the average sum of the logarithm of the scaling factors converges with probability one to the entropy rate; thus the variance σ_∞^2 is zero. For general Markov processes, this may not be true and there is a variance in the limit that is strictly bigger than zero (see Chen [21]).

3.3 Computing Lower Bounds on Capacity

In absence of noise, the capacity of an FSM depends only on the maximum number of different sequences it generates. This number is independent of a particular "encoder" realization, and thus artificially increasing the state-space of the FSM beyond its minimal state-space realization does not increase capacity.

In the presence of noise, this is different because the mutual information between the FSM input X^n, or equivalently the FSM state sequence S^n, and the output Y^n can be expanded as follows:

$$\begin{aligned} I(X^n;Y^n) &= H(X^n) - H(X^n|Y^n) && (3.78)\\ &= H(X^n) - \sum_{i=1}^{n} H(X_i|X_1^{i-1},Y^n) && (3.79)\\ &\geq H(X^n) - \sum_{i=1}^{n} H(X_i|X_{i-1},Y^n)\,. && (3.80) \end{aligned}$$

In case of Gaussian inputs, this is exploited by the "water-filling" technique [42]. The "water-filling" technique aims at increasing the SNR at the channel output by shaping the input spectrum to the channel spectrum. The input bits are no longer i.i.d. but correlated and their spectrum is proportional to the inverse of the channel spectrum (thus the name "water-filling").

As mentioned in Chapter 2, we assume Markov sources as input processes because they are easily describable. For a given order of the Markov input process, the state-transition probabilities are free parameters that can be optimized in order to maximize the SNR at the channel output which in turn will increase the information rate. Additional free parameters are obtained by artificially increasing the Markov model of the source beyond its minimal state-space realization. By doing so, the rigid structure of the Markov source is relaxed.

Example 3.3 (Extension of Bernoulli Processes)
If the input signals are i.i.d. distributed, the input process is a Bernoulli process *and the minimal state-space realization requires only one state. The first extension of a Bernoulli process is a memory-one (two states) Markov process, and the second extension is a memory-two (four states) Markov process.*

Example 3.4 (Extension of the $(1,\infty)$-RLL Source)
As mentioned in Chapter 2, the $(1,\infty)$-RLL constraint can be realized by a two state Markov source. This is the minimal state-space realization. The first extension has three and second extension five states.

3.3.1 FSM Information Rate and FSM Capacity

We start by defining for a given extension η of the input process the FSM information rate. The minimal state-space realization is always denoted by $\eta = 0$.

Definition 3.4 (FSM Information Rate) *For a given extension η of the input process, and a given set of state-transition probabilities Q of that process, the FSM information rate is defined as*

$$I_\eta(\mathsf{Q}) \triangleq -\sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j)\left[\log_2 \mathbf{Q}(i,j) - \mathbf{T}(i,j)\right] \tag{3.81}$$

$$= \sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j)\log_2 \frac{\tilde{\mathbf{A}}(i,j)}{\mathbf{Q}(i,j)}. \tag{3.82}$$

We continue and define the *FSM capacity* C_η as the maximal FSM information rate achievable when the Markov input process is the η-th extension of the input process.

Definition 3.5 (FSM Capacity) *Let the η-th extension of the Markov input process be determined by the set of state-transition probabilities Q. For a fixed extension η of the Markov input process, the FSM capacity is defined by*

$$C_\eta^{\mathrm{FSM}} \triangleq \max_{\mathsf{Q}} I_\eta(\mathsf{Q}) \tag{3.83}$$

$$= \max_{\mathsf{Q}} \left[- \sum_{\substack{(i,j) \\ \mathbf{A}(i,j)=1}} \boldsymbol{\pi}(i)\mathbf{Q}(i,j) \left[\log_2 \mathbf{Q}(i,j) - \mathbf{T}(i,j)\right] \right] \tag{3.84}$$

$$= - \sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}^*(i)\mathbf{Q}^*(i,j) \left[\log_2 \mathbf{Q}^*(i,j) - \mathbf{T}^*(i,j)\right] \tag{3.85}$$

$$= \sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}^*(i)\mathbf{Q}^*(i,j) \log_2 \frac{\tilde{\mathbf{A}}^*(i,j)}{\mathbf{Q}^*(i,j)} \tag{3.86}$$

where the maximization is performed over all sets of state-transition probabilities Q that form a valid set for the η-th extension of the input process. The set of state-transition probabilities that achieve the FSM capacity *are denoted by Q^*. Note that in (3.86), the probabilities $\boldsymbol{\pi}^*(i)$ and $\mathbf{Q}^*(i,j)$ are uniquely determined by Q^* and that $\mathbf{T}^*(i,j)$ (and thus $\tilde{\mathbf{A}}^*(i,j)$) depends on Q^* as given in (3.55) and (3.70) respectively.*

Thus, a sequence of lower bounds on the capacity of the FSC is obtained in the following way

$$C_0^{\mathrm{FSM}} \leq C_1^{\mathrm{FSM}} \leq \ldots \leq C_\infty^{\mathrm{FSM}} = C^{\mathrm{FSC}} \tag{3.87}$$

which in the limit (for an infinite extension of the input process) equals the FSC channel capacity.

3.3.2 Lagrangian and Gradient-Based Search

Computing the FSM capacity requires a maximization of the state-transition probabilities of the source. This is complicated by the fact that analytical expressions for the information rate itself and for its derivatives in function of Q are very complicated.

Our objective-function is the Lagrangian consisting of the FSM information rate and the constraints which are imposed on the Markov input source. Thus we want to maximize the

Lagrangian

$$\begin{aligned}\mathcal{L} &= I_\eta(\mathsf{Q}) + \sum_{i\in\mathsf{S}} \lambda_i \sum_{j\in\mathsf{S}} \mathbf{Q}(i,j) \\ &\quad + \sum_{j\in\mathsf{S}} \lambda'_j \left(\sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i)\mathbf{Q}(i,j) - \boldsymbol{\pi}(j) \right) + \lambda'' \sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i) \qquad (3.88) \\ &= \sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j) \log_2 \frac{\tilde{\mathbf{A}}(i,j)}{\mathbf{Q}(i,j)} \\ &\quad + \sum_{i\in\mathsf{S}} \lambda_i \sum_{j\in\mathsf{S}} \mathbf{Q}(i,j) + \sum_{j\in\mathsf{S}} \lambda'_j \left(\sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i)\mathbf{Q}(i,j) - \boldsymbol{\pi}(j) \right) \\ &\quad + \lambda'' \sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i) \qquad (3.89)\end{aligned}$$

over Q, where all entries $\tilde{\mathbf{A}}(i,j)$ of the noisy adjacency matrix depend on the set of state-transition probabilities Q.

Gradient-Based Search

Brute force optimization of the state-transition probabilities is only a practical approach for very small input processes. For a higher order input process, iterative gradient techniques can be used. The starting point is a randomly selected set of state-transition probabilities $\mathsf{Q}^{(0)}$ that fulfills the Markov constraints. The update equation for a particular state-transition probability $\mathbf{Q}^{(\ell+1)}(i,j)$ at the $(\ell+1)$-th iteration can be given in the following form

$$\mathbf{Q}^{(\ell+1)}(i,j) = \mathbf{Q}^{(\ell)}(i,j) + \varsigma \frac{I_\eta\left(\mathbf{Q}^{(\ell)}(i,j)\right) - I_\eta\left(\mathbf{Q}^{(\ell)}(i,j) + \boldsymbol{\Delta}(i,j)\right)}{\boldsymbol{\Delta}(i,j)}. \qquad (3.90)$$

For finite-block lengths, the evaluation of the FSM information rate (using the forward-only method) is reliable only within $O(n^{-\frac{1}{2}})$. Assuming that the information rate is concave in $\mathbf{Q}^{(\ell)}(i,j)$, $\boldsymbol{\Delta}(i,j)$ must be roughly twice as large as this tolerance; otherwise the numerator in (3.90) is a random variable. The update equation (3.90) is most efficiently implemented in a *zooming* way: First short block-lengths and large $\boldsymbol{\Delta}(i,j)$ are used and afterwards long block-lengths with small $\boldsymbol{\Delta}(i,j)$ so as to increase the precision from iteration to iteration. The step size ς is chosen to be sufficiently small, but may depend on the state-transition and can be decreased in a similar way as $\boldsymbol{\Delta}(i,j)$ is decreased.

3.3.3 Markov Constrained Arimoto-Blahut Algorithm

We construct the Arimoto-Blahut algorithm under the constraint that the source is a Markov process. To construct this algorithm, we need to define a skewed a-posteriori state-transition

weight matrix $\mathbf{t}$ with entries $\mathbf{t}(i,j)$ for every state-transition $(i,j) : \mathbf{A}(i,j) = 1$.

Definition 3.6 (Skewed A-posteriori State-Transition Weight) *For all transitions* $(i,j) : \mathbf{A}(i,j) = 1$*, we define the skewed a-posteriori state transition weight in the following way:*

$$\mathbf{t}^{(\ell)}(i,j) \triangleq \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \mathrm{E}_{r_{Y^n}} \left[\log_2 \frac{v_k^{(\mathsf{Q}^{(\ell)})}(i,j|Y^n)^{\frac{v_k^{(\mathsf{Q})}(i,j|Y^n)}{\boldsymbol{\pi}(i)\mathbf{Q}(i,j)}}}{v_{k-1}^{(\mathsf{Q}^{(\ell)})}(i|Y^n)^{\frac{v_{k-1}^{(\mathsf{Q})}(i|Y^n)}{\boldsymbol{\pi}(i)}}} \right] \tag{3.91}$$

otherwise $\mathbf{t}^{(\ell)}(i,j) = -\infty$*. The output pdf* $r_{Y^n}(\cdot)$ *is a function of the set of STPs* Q*. Note that the backward channel laws in the exponent of numerator and denominator depend on a different set of STPs, namely on* Q *instead of* $\mathsf{Q}^{(\ell)}$ *(where* $\ell = 0, 1, 2, \ldots$ *is an index).*

Note that substituting $\mathbf{t}^{(\ell)}(i,j)$ for $\mathbf{T}(i,j)$ in (3.81) does not provide an "FSM information rate" unless $\mathbf{t}^{(\ell)}(i,j) = \mathbf{T}(i,j)$. This is the case if $\mathsf{Q}^{(\ell)} = \mathsf{Q}$.

An estimate of the value $\mathbf{t}(i,j)$ can be computed using the sum-product algorithm with *two* long output sequences y^n (one for the source Q and one for the source $\mathsf{Q}^{(\ell)}$) , i.e.

$$\hat{\mathbf{t}}^{(\ell)}(i,j) = \frac{1}{n} \sum_{k=1}^{n} \log_2 \frac{v_k^{(\mathsf{Q}^{(\ell)})}(i,j|y^n)^{\frac{v_k^{(\mathsf{Q})}(i,j|y^n)}{\boldsymbol{\pi}(i)\mathbf{Q}(i,j)}}}{v_{k-1}^{(\mathsf{Q}^{(\ell)})}(i|y^n)^{\frac{v_{k-1}^{(\mathsf{Q})}(i|y^n)}{\boldsymbol{\pi}(i)}}} \ . \tag{3.92}$$

For n large, $\hat{\mathbf{t}}(i,j)$ converges to $\mathbf{t}(i,j)$ with probability 1.

We can now formulate the Markov-constrained Arimoto-Blahut algorithm.

Algorithm 3.6 Markov-Constrained Arimoto-Blahut Algorithm

Step 0: *Initialize the algorithm with an arbitrary set of STPs* $\mathsf{Q}^{(0)}$*.*

Step 1: *Find* $\mathsf{Q}^{(\ell+1)}$ *as*

$$\mathsf{Q}^{(\ell+1)} = \underset{\mathsf{Q}}{\operatorname{argmax}} \left[- \sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j) \left[\log_2 \mathbf{Q}(i,j) - \mathbf{t}^{(\ell)}(i,j) \right] \right] . \tag{3.93}$$

Increment ℓ *by* 1*, and repeat Step 1 until convergence. Note that* $\mathbf{t}^{(\ell)}(i,j)$ *depends on both* Q *and* $\mathsf{Q}^{(\ell)}$ *but the probabilities* $\boldsymbol{\pi}(i)$ *and* $\mathbf{Q}(i,j)$ *are uniquely determined by* Q*.*

An obvious difficulty with the Markov-constrained Arimoto-Blahut algorithm is that the algorithm is computationally extremely complex to execute. This is because we need to run the sum-product algorithm at every point Q in order to numerically evaluate $\mathbf{t}^{(\ell)}(i,j)$, which we

need in order to evaluate the maximum in Step 1. This is clearly computationally prohibitive. On the other hand, this algorithm provably converges to a local maximum. Moreover, if $I_\eta(\mathsf{Q})$ is concave in Q (which we actually don't know), this algorithm converges to C_η^{FSM}.

From (3.87) follows that for an infinite extension of the input process, we obtain a capacity-achieving input distribution of the FSC (although this distribution may be impossible to compute in practice).

3.3.4 Iterative Information Rate Maximization Method

The following elegant algorithm to maximize the information rate is due to Kavčić [47]. It is an iterative algorithm that is initialized with an arbitrary choice of STPs. Steps 0 and 1 are performed iteratively.

Algorithm 3.7 Iterative Information Rate Maximization Method [47]

Step 0: *Given* $\mathsf{Q}^{(\ell)}$*, evaluate* $\mathbf{T}^{(\ell)}(i,j)$*. We write* $\mathbf{T}^{(\ell)}(i,j)$ *instead of* $\mathbf{T}(i,j)$ *to emphasize that the a-posteriori state-transition weight is a function of* $\mathsf{Q}^{(\ell)}$ *and* not *of* Q*.*

Step 1: *Find* $\mathsf{Q}^{(\ell+1)}$ *as*

$$\mathsf{Q}^{(\ell+1)} = \operatorname*{argmax}_{\mathsf{Q}} \left[- \sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}(i,j) \left[\log_2 \mathbf{Q}(i,j) - \mathbf{T}^{(\ell)}(i,j) \right] \right]. \tag{3.94}$$

Increment ℓ *by one and go back to Step 0. In (3.94), the probabilities* $\boldsymbol{\pi}(i)$ *and* $\mathbf{Q}(i,j)$ *are determined by* Q*.*

Step 0 is performed by running the forward-backward sum-product algorithm and numerically evaluating estimates of $\mathbf{T}^{(\ell)}(i,j)$ using (3.58).

Theorem 3.5 *Let $\rho(\tilde{\mathbf{A}})$ denote the Perron-root of the noisy adjacency matrix $\tilde{\mathbf{A}}$, and let $\tilde{\boldsymbol{r}}^T$ be the right Perron-vector.*

a) With (3.94) we obtain the following information rate

$$\begin{aligned} I_\eta^{(\ell)}(\mathsf{Q}^{(\ell+1)}) &\triangleq \max_{\mathsf{Q}} \left[\sum_{\substack{(i,j) \\ \mathbf{A}(i,j)=1}} \boldsymbol{\pi}(i)\mathbf{Q}(i,j) \left[\log_2 \frac{1}{\mathbf{Q}(i,j)} + \mathbf{T}^{(\ell)}(i,j) \right] \right] \\ &= \sum_{(i,j):\mathbf{A}(i,j)=1} \boldsymbol{\pi}(i)\mathbf{Q}^{(\ell+1)}(i,j) \left[\log_2 \frac{1}{\mathbf{Q}^{(\ell+1)}(i,j)} + \mathbf{T}^{(\ell)}(i,j) \right] \\ &= \log_2 \rho(\tilde{\mathbf{A}}) \end{aligned} \tag{3.95}$$

The superscript ℓ in $I_\eta^{(\ell)}(\mathsf{Q}^{(\ell+1)})$ reminds that $\mathbf{T}^{(\ell)}(i,j)$ depends on $\mathsf{Q}^{(\ell)}$.

b) The set of state-transition probabilities $\mathsf{Q}^{(\ell+1)}$ that achieves the maximization in (3.94) contains the following individual state-transition probabilities

$$\mathbf{Q}(i,j)^{(\ell+1)} = \frac{\tilde{\boldsymbol{r}}(j)}{\tilde{\boldsymbol{r}}(i)} \cdot \frac{2^{\mathbf{T}^{(\ell)}(i,j)}}{\rho(\tilde{\mathbf{A}})}. \tag{3.96}$$

The right Perron-vector $\tilde{\boldsymbol{r}}^T$ originates from the noisy adjacency matrix $\tilde{\mathbf{A}}$ with entries $\tilde{\mathbf{A}}(i,j) = 2^{\mathbf{T}^{(\ell)}(i,j)}$.

Proof: A proof can be found in Appendix C. ■

From (3.96), we know that

$$I_\eta^{(\ell)}(\mathsf{Q}^{(\ell)}) \leq I_\eta^{(\ell)}(\mathsf{Q}^{(\ell+1)}) \,. \tag{3.97}$$

If

$$I_\eta^{(\ell)}(\mathsf{Q}^{(\ell+1)}) \leq I_\eta^{(\ell+1)}(\mathsf{Q}^{(\ell+1)}) \tag{3.98}$$

also holds, then

$$I_\eta^{(1)}(\mathsf{Q}^{(1)}) \leq I_\eta^{(1)}(\mathsf{Q}^{(2)}) \leq I_\eta^{(2)}(\mathsf{Q}^{(2)}) \leq \cdots \leq I_\eta^{(\infty)}(\mathsf{Q}^{(\infty)}) = C_\eta^{\mathrm{FSM}}. \tag{3.99}$$

Conjecture 3.6 (Kavčić Conjecture [47]) *For a given order of the input process, the iterative information maximization method increases the information rate and converges (at least) to a local maximum.*

Note the iterative information maximization method is *not* an expectation-maximization algorithm because each update of the state-transition probabilities operates on a *new* sequence y^n. This algorithm and its relation to the Arimoto-Blahut algorithm have been investigated by Vontobel in [74].

3.4 Computing Upper Bounds on Capacity

The idea of the following upper bound is based on the information-theoretic fact that although there are in general several capacity-achieving input distributions, the corresponding output distribution is unique (for a proof see (2.80)). The tightness of the upper bound depends on how close the output distribution is to the unique output distribution. The weakness of the following upper bound is that it does not take into account the Markov property of the input process. Thus it is also an upper bound on the capacity of the FSC.

Consider a DMC with input X and output Y. From Chapter 2, we know that capacity can be written as a weighted divergence between the channel law $w(y|x)$ and the output distribution $r^*(\cdot)$ induced by $q^*(\cdot)$ as

$$C = \sum_x q^*(x) D\left(w(\cdot|x)||r^*(\cdot)\right) \tag{3.100}$$

where the summation of the divergence goes over y for every x. The divergence term can also be written as in (2.77). In order to achieve capacity, the divergence terms, must be equal for all x with probability greater than zero (Karush-Kuhn-Tucker condition). Hence, we have

$$\begin{aligned} C &= \min_{r(\cdot)} \max_x \sum_x q(x) D\left(w(\cdot|x)||r(\cdot)\right) && (3.101)\\ &= D\left(w(\cdot|x)||r^*(\cdot)\right) \quad \forall\, x \in \mathsf{X} && (3.102)\\ &\leq \max_x D\left(w(\cdot|x)||r(\cdot)\right) \quad \text{for any } r(\cdot) \text{ over } \mathsf{Y}. && (3.103) \end{aligned}$$

The inequality in the last line follows from the fact that capacity is a strictly convex function of the output distribution $r(\cdot)$ (see also (2.78)).

Generalizing the upper bound to sequences, the capacity of an indecomposable FSC and therefore of an irreducible and aperiodic FSM with memory $m > 0$ can be upper bounded by

$$C \leq \lim_{n\to\infty} \max_{x^n} \frac{1}{n} D\left(w(\cdot|x_{1-m}^n)||r(\cdot)\right), \tag{3.104}$$

where the maximization is over all possible branch sequences x^n of length n starting at time zero in a well-defined state x_{1-m}^0. Again, we assume here that the input sequence induces an unique branch sequence through the FSM. We will call this maximizing branch sequence the "worst-case" branch sequence. The output distribution $r(\cdot)$ may be *any* n-dimensional probability distribution. If we assume additive white noise, the channel law can be factored in the usual way, i.e.

$$w(y^n|x_{1-m}^n) = \prod_{k=1}^n w(y_k|x_{k-m}^k). \tag{3.105}$$

As we are free in choosing $r(\cdot)$, the approach here is to model/approximate the output as a Markov process of order $M \geq m$, i.e.

$$r(y^n) = \prod_{k=1}^{n} r(y_k | y_{k-M}^{k-1}). \tag{3.106}$$

This approximation is crucial because it allows us to rebuild the output at time k from a *finite* number $M < \infty$ of previous outputs, i.e. to predict y_k from M past output samples. In other words, the output distribution of the hidden Markov model that in general is not Markov, i.e. $M = \infty$, is approximated by an FSM of order $L = 2^M$ (we consider binary input signals). It has been realized by Vontobel [75], that this allows us to compute the divergence term $D\left(w(\cdot|x_{1-m}^n)||r(\cdot)\right)$ in a similar way as the output entropy rate $h(\mathcal{Y})$ with probability one for $n \to \infty$.

By definition, the divergence term is

$$D\left(w(\cdot|x_{1-m}^n)||r(\cdot)\right) = \sum_{y^n} w(y^n|x_{1-m}^n) \log_2 \frac{w(y^n|x_{1-m}^n)}{r(y^n)}. \tag{3.107}$$

The numerator is rewritten with the help of (3.105) as follows

$$\begin{aligned}
\sum_{y^n} w(y^n|x_{1-m}^n) \log_2 w(y^n|x_{1-m}^n) &= \mathrm{E}_{w_{Y^n|X_{1-m}^n = x_{1-m}^n}} \left[\log_2 w(y^n|x_{1-m}^n)\right] && (3.108)\\
&= \mathrm{E}_{w_{Y^n|X_{1-m}^n = x_{1-m}^n}} \left[\sum_{k=1}^{n} \log_2 w(y_k|x_{k-m}^k)\right] && (3.109)\\
&= \sum_{k=1}^{n} \mathrm{E}_{w_{Y_k|X_{k-m}^k = x_{k-m}^k}} \left[\log_2 w(y_k|x_{k-m}^k)\right] && (3.110)\\
&= -\sum_{k=1}^{n} H(Z_k) && (3.111)\\
&= -H(Z_1) && (3.112)
\end{aligned}$$

where the last line follows from the assumption that the statistics of the additive noise is independent of time. The denominator in (3.107) is expanded with the help of (3.106) in the following way

$$\begin{aligned}
\sum_{y^n} w(y^n|x^n) \log_2 r(y^n) &= \mathrm{E}_{w_{Y^n|X_{1-m}^n = x_{1-m}^n}} \left[\log_2 r(y^n)\right] && (3.113)\\
&= \mathrm{E}_{w_{Y^n|X_{1-m}^n = x_{1-m}^n}} \left[\sum_{k=1}^{n} \log_2 r(y_k|y_{k-M}^{k-1})\right] && (3.114)\\
&= \sum_{k=1}^{n} \mathrm{E}_{w_{Y_{k-M}^k|X_{k-(M+m)}^k = x_{k-(M+m)}^k}} \left[\log_2 r(y_k|y_{k-M}^{k-1})\right] && (3.115)
\end{aligned}$$

We introduce now the state $\tilde{s}_{k-1} = x_{k-(M+m)}^{k-1}$ and express (3.107) with the help of (3.112) with (3.115) through the following definitions

$$f(x_{k-M+m}^{k}) \triangleq \tilde{f}(\tilde{s}_{k-1}, x_k) \tag{3.116}$$

$$\triangleq -H(Z_1) - \mathrm{E}_{w_{Y_{k-M}^{k}|X_{k-(M+m)}^{k}=x_{k-(M+m)}^{k}}}\big[\log_2 r(\cdot|y_{k-M}^{k-1})\big]. \tag{3.117}$$

If the noise is not discrete (as we assume here) the summation in (3.107) is replaced by an integration and the entropy term $H(Z_1)$ in (3.112) and (3.117) by the differential entropy term $h(Z_1)$.

Up to now, we used $r(\cdot)$ without specifying how it is expressed in terms of the channel output distribution. To do so, we define

$$r(y_k|y_{k-M}^{k-1}) \triangleq \frac{\tilde{r}(y_{k-M}^{k})}{\sum_{y_k} \tilde{r}(y_{k-M}^{k})} \tag{3.118}$$

such that

$$\begin{aligned}\tilde{f}(\tilde{s}_{k-1}, x_k) = -H(Z_1) &- \mathrm{E}_{w_{Y_{k-M}^{k}|X_{k-(M+m)}^{k}=x_{k-(M+m)}^{k}}}\big[\log_2 \tilde{r}(y_{k-M}^{k})\big] \\ &+ \mathrm{E}_{w_{Y_{k-M}^{k}|X_{k-(M+m)}^{k}=x_{k-(M+m)}^{k}}}\big[\log_2 \tilde{r}(y_{k-M}^{k-1})\big].\end{aligned} \tag{3.119}$$

For $\tilde{r}(\cdot)$, we can choose any multi-dimensional distribution, but the idea is of course to use (for sake of simplicity) the output distribution of the original joint source/channel model with memory m for which we aim at computing the upper bound. As there are 2^{M+1} different branches (we assume binary inputs) of length $M+1$ starting in any of the 2^m different states of the joint/source channel model, we have $2^m \cdot 2^{M+1}$ different values $\tilde{r}(y_{k-M}^{k})$ to compute. The computation of these values can be done by starting in a specific state, collecting an output sequence of length $M+1$, and performing the sum-product forward recursion. Since the branch length $M+1$ can not be too big — otherwise the number of different branches of length $M+1$ is formidable —, several such computations are performed to approximate the expectations in (3.119) by stochastic averaging (law of large numbers).

From (2.80), we know that the closer $\tilde{r}(\cdot)$ is to $r^*(\cdot)$, the tighter the upper bound becomes. This can be achieved simply by inceasing M. Moreover, we can tighten the upper bound for a given M by using $\tilde{r}(y_{k-M}^{k})$s that orginate from a tight lower bound. Thus, we first maximize the FSM information rate to obtain a tight lower bound and the corresponding SNR dependent STPs. Afterwards, $\tilde{r}(y_{k-M}^{k})$ is computed with these optimized STPs. Hence, an extended input process leads to tighter lower bounds as well as tighter upper bounds.

If the channel is time-independent (as we assume), we have $f(x_{k-M}^{k}) = f(x_1^{M+1}) = \tilde{f}(\tilde{s}_{k-1}, x_k) = \tilde{f}(\tilde{s}_M, x_{M+1})$ for all $M+1 \leq k \leq n$ (trellis initializing phase is excluded), independently of the current value of k. There are 2^{m+M+1} different $\tilde{f}(\tilde{s}_{k-1}, x_k)$ which can

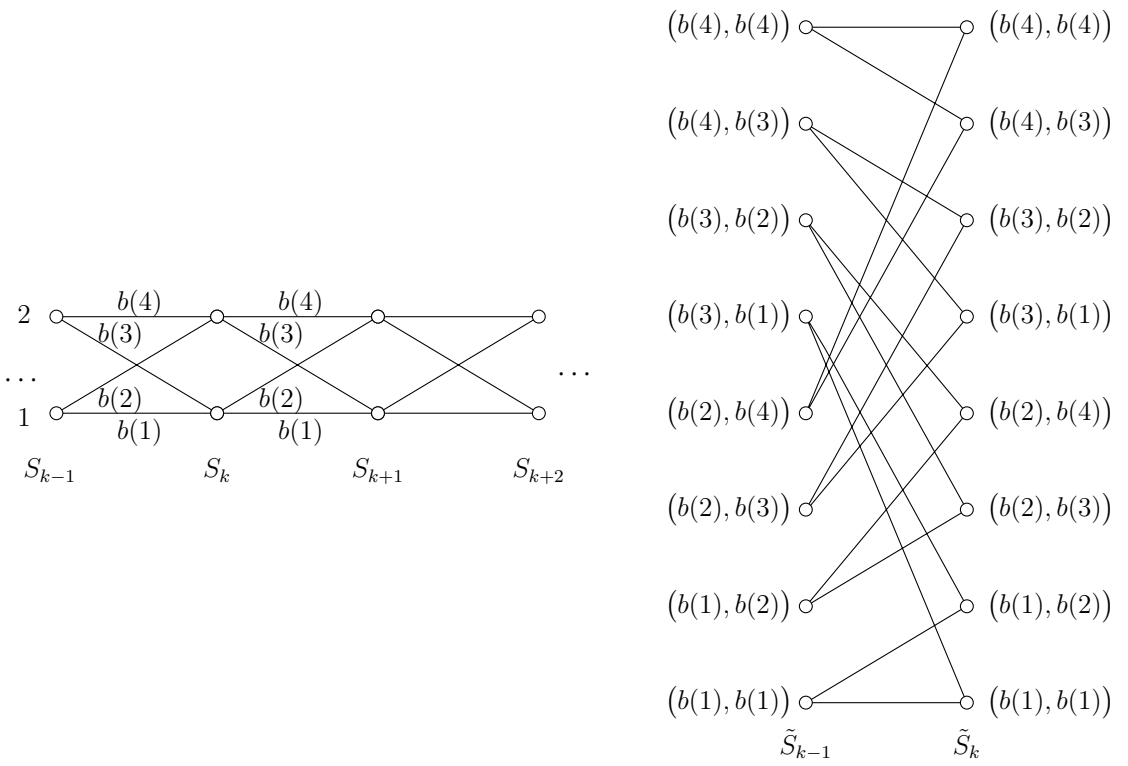

Figure 3.1: Three trellis sections of the DICODE channel (left part) with $m = 1$ and one trellis section of new trellis (right part) with $M = 2$.

be arranged in a *new* trellis with $2^M \cdot 2^m$ different states. In this new trellis, there are two branches leaving from each state as the input of the original trellis is binary. The state $\tilde{s}_{k-1} = x_{k-M}^{k-1} = b_{k-M}^{k-1}$ at time $k-1$ is connected to a state $\tilde{s}_k = x_{k-(M-1)}^{k} = b_{k-(M-1)}^{k}$ at time k with a branch to which the metric $\tilde{f}(\tilde{s}_{k-1}, x_k)$ is assigned.

Example 3.5 (DICODE Channel)
This procedure is visualized in Fig. 3.1. The left part shows three trellis sections of the DICODE channel ($m = 1$). The branches of the first two trellis sections are labelled, as we assume in this example $M = 2$. Because the channel is time-invariant, the branches in these two trellis sections (transition from S_{k-1} to S_k and from S_k to S_{k+1}) carry the same label (provided that the initialization phase is over). There are $2^M = 4$ possible sequences of length $M = 2$ from state $S_{k-1} = 1$ to a time-$(k+1)$ state. The same holds for $S_{k-1} = 2$. We can form eight different states, $\tilde{S}_{k-1}$, in the new trellis by the branches of the old trellis, i.e.

$$\tilde{S}_{k-1} = \big(b(i), b(j)\big)\,,$$

as eight out of 16 combinations are feasible. The input is binary and consequently there are 16 branches in the new trellis.

The trellis sections in the new trellis become identical after m time steps the latest where m is the memory of the joint source/channel model. For a specific x_{1-m}^{n}, $D\left(w(\cdot|x_{1-m}^{n})||r(\cdot)\right)$ equals the cumulative metric along the corresponding path in the new trellis. We neglect the

initial m trellis sections in the new trellis, as they are not relevant for the calculation of the upper bound. In the remaining part of the trellis, all sections are identical and independent of the length n. Finding the "worst-case" sequence x_{1-m}^{n} for $n \to \infty$ is now equivalent to finding the semi-infinite path x^{∞} starting at time zero in this new trellis with the largest metric or equivalently with the largest metric $\tilde{f}(\tilde{s}_{k-1}, x_k)$ per trellis section.

If the worst case sequence is unique it must be periodic with a period length not larger than 2^{m+M}. This is because in the new trellis all sections and all branch metrics are equal (after the first m trellis sections). Hence, being in a specific state at any time, one always takes the same branch from this state to travel to a state in the next time step. As there are 2^{m+M} states, we must arrive after at most 2^{m+M} time steps in the same state again. To determine this largest metric per trellis section we can use a modified Viterbi (max-sum) algorithm. If there are several paths having the same largest metric, there is also a periodic one among them. As we are only interested in the metric per trellis section and not the path itself, we can simply search for this periodic "worst-case" sequence.

3.5 Information Rates of General Channels

The methods presented in the first section of this chapter can be extended to compute upper and lower bounds on the information rate of very general (non-finite-state) channels. For the sake of clarity, we begin by stating the bounds for the discrete memoryless case. Let X be the input and O be the *observed* output of a discrete memoryless channel, termed *original channel*, with joint pmf $p(x, o)$ (see Fig. 3.2). Let the channel law of the original channel be $w(o|x)$. We define the information rate of the original channel in the usual way, i.e.

$$I(X;O) \triangleq \sum_{x,o} p(x,o) \log_2 \frac{p(x,o)}{q(x)r(o)}. \tag{3.120}$$

Let $w_{\mathrm{a}}(y|x)$ be the law of an arbitrary *auxiliary-channel* with the same input and output alphabets as the original channel. The auxiliary-channel is connected to the same source X (see Fig. 3.2); its output Y is then distributed according to

$$r_{\mathrm{a}}(y) \triangleq \sum_{x} q(x)\, w_{\mathrm{a}}(y|x). \tag{3.121}$$

3.5.1 Upper and Lower Bounds

We start by defining an upper bound on the information rate $I(X;O)$.

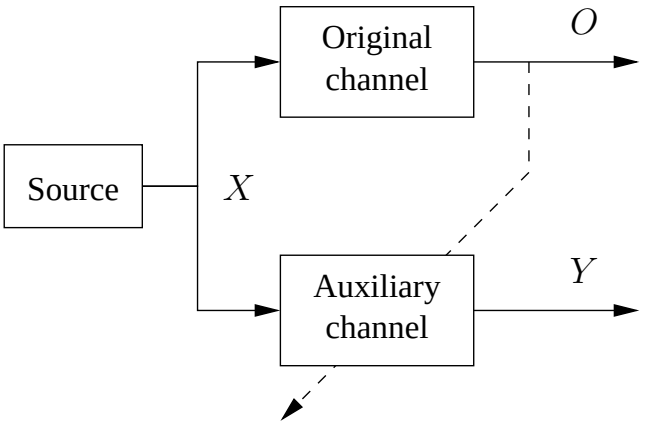

Figure 3.2: System overview.

Definition 3.7 (Upper Bound)

$$\overline{I}_{\mathrm{a}}(X;O) \triangleq \sum_{x,o} p(x,o) \log_2 \frac{w(o|x)}{r_{\mathrm{a}}(o)} \tag{3.122}$$

$$= E_{p_{X,O}} \left[\log_2 w(O|X) - \log_2 r_{\mathrm{a}}(O) \right]. \tag{3.123}$$

Theorem 3.7 (Auxiliary-Channel Upper Bound)

$$\overline{I}_{\mathrm{a}}(X;O) \geq I(X;O). \tag{3.124}$$

Proof: This bound appears to have been observed first by Topsøe [73] (see also Section 2.3). The proof is equivalent to (2.80) and is given here for the sake of completeness.

$$\overline{I}_{\mathrm{a}}(X;O) - I(X;O) = \sum_{x,o} p(x,o) \left[\log_2 \frac{w(o|x)}{r_{\mathrm{a}}(o)} - \log_2 \frac{w(o|x)}{r(o)} \right]$$

$$= \sum_{x,o} p(x,o) \log_2 \frac{r(o)}{r_{\mathrm{a}}(o)} \tag{3.125}$$

$$= \sum_{o} r(o) \log_2 \frac{r(o)}{r_{\mathrm{a}}(o)} \tag{3.126}$$

$$= D\big(r(\cdot)||r_{\mathrm{a}}(\cdot)\big) \tag{3.127}$$

$$\geq 0. \tag{3.128}$$

■

The lower bound is implicit in the classical papers by Arimoto [1] and Blahut [18].

Definition 3.8 (Lower Bound)

$$\underline{I}_{\mathrm{a}}(X;O) \triangleq \sum_{x,o} p(x,o) \log_2 \frac{w_{\mathrm{a}}(o|x)}{r_{\mathrm{a}}(o)} \tag{3.129}$$

$$= E_{p_{X,O}} \left[\log_2 w_{\mathrm{a}}(O|X) - \log_2 r_{\mathrm{a}}(O) \right]. \tag{3.130}$$

Theorem 3.8 (Auxiliary-Channel Lower Bound)

$$I(X;O) \geq \underline{I}_{\mathrm{a}}(X;O). \tag{3.131}$$

Proof: Let

$$v_{\mathrm{a}}(x|o) \triangleq \frac{q(x) w_{\mathrm{a}}(o|x)}{r_{\mathrm{a}}(o)} \tag{3.132}$$

be the backward channel law of the auxiliary-channel. Then

$$I(X;O) - \underline{I}_{\mathrm{a}}(X;O) = \sum_{x,o} p(x,o) \left[\log_2 \frac{p(x,o)}{q(x) r(o)} - \log_2 \frac{w_{\mathrm{a}}(o|x)}{r_{\mathrm{a}}(o)} \right]$$

$$= \sum_{x,o} p(x,o) \log_2 \frac{p(x,o)}{r(o) q(x) w_{\mathrm{a}}(o|x) / r_{\mathrm{a}}(o)} \tag{3.133}$$

$$= \sum_{x,o} p(x,o) \log_2 \frac{p(x,o)}{r(o) v_{\mathrm{a}}(x|o)} \tag{3.134}$$

$$= D\big(p(\cdot,\cdot) || r(\cdot) v_{\mathrm{a}}(\cdot|\cdot)\big) \tag{3.135}$$

$$\geq 0. \tag{3.136}$$

■

This lower bound may be obtained as a special case of a bound due to Fischer [31] on mismatched decoding, which in turn is a special case of the general result by Ganti at al. [35, Equation (12) for $s = 1$]. It then follows from the results in [31] and [35] that the lower bound is achievable by a maximum-likelihood decoder for the auxiliary-channel.

From the proofs follows that both the upper bound (3.124) and the lower bound (3.131) are tight if and only if $q(x) w_{\mathrm{a}}(o|x) = p(x,o)$ for all x and o. The difference between the upper

and the lower bound can be expressed as follows:

$$\begin{aligned}
\overline{I}_{\mathrm{a}}(X;O) - \underline{I}_{\mathrm{a}}(X;O) &= \sum_{x,o} p(x,o) \left[\log_2 \frac{w(o|x)}{r_{\mathrm{a}}(o)} - \log_2 \frac{w_{\mathrm{a}}(o|x)}{r_{\mathrm{a}}(o)} \right] \\
&= \sum_{x,o} p(x,o) \left[\log_2 \frac{w(o|x)}{w_{\mathrm{a}}(o|x)} \right] && (3.137) \\
&= \sum_{x} q(x) \sum_{o} w(o|x) \log_2 \frac{w(o|x)}{w_{\mathrm{a}}(o|x)} && (3.138) \\
&= \sum_{x} q(x) D\big(w(\cdot|x) || w_{\mathrm{a}}(\cdot|x)\big) && (3.139) \\
&\geq 0. && (3.140)
\end{aligned}$$

If the auxiliary-channel is a parameterized model, the difference between the upper and lower bound can be decreased by adjusting its model parameters such that the divergence terms in (3.139) becomes small. This *training* of the auxiliary-channel is visualized with dotted lines in Fig. 3.2.

3.5.2 Generalization to Channels with Memory

The generalization of these bounds to the information rate of an indecomposable finite-state channel as original channel and an irreducible and aperiodic FSM as auxiliary channel is straightforward. The upper bound becomes

$$\overline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O}) \triangleq \lim_{n\to\infty} \frac{1}{n} \mathrm{E}_{p_{X^n,O^n}} \left[\log_2 w(O^n|X_{1-M}^n) - \log_2 r_{\mathrm{a}}(O^n) \right] \tag{3.141}$$

where $M \geq 1$ is the memory of the original channel. The lower bound becomes

$$\underline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O}) \triangleq \lim_{n\to\infty} \frac{1}{n} \mathrm{E}_{p_{X^n,O^n}} \left[-\log_2 r_{\mathrm{a}}(O^n) + \log_2 w_{\mathrm{a}}(O^n|X_{1-m}^n) \right] \tag{3.142}$$

with $m \geq 1$ as the memory of the auxiliary channel. Now assume that $w(\cdot|\cdot)$ is some "difficult" (non-finite-state) ergodic channel. Its information rate is defined as follows

$$I(\mathcal{X};\mathcal{O}) \triangleq \lim_{n\to\infty} \frac{1}{n} \mathrm{E}_{p_{X^n,O^n}} \left[-\log_2 r(O^n) + \log_2 w(O^n|X_{1-M}^n) \right]. \tag{3.143}$$

We can then compute estimates of upper and lower bounds on the information rate of the general (non-finite-state) channel by the following algorithm:

Algorithm 3.8 Bounds on the Information Rate by Means of an FSM

Step 0: *Choose a finite-state source* Q *and an auxiliary finite-state channel* $w_{\mathrm{a}}(\cdot|\cdot)$ *so that their concatenation is an FSM.*

Step 1: *Concatenate the source to the* original *channel* $w(\cdot|\cdot)$ *and generate two long sequences* x^n *and* o^n*.*

Step 2: *Compute* $\log_2 r_{\mathrm{a}}(o^n)$ *and* $\log_2 w_{\mathrm{a}}(o^n|x_{1-m}^n)$ *by the method described in the first section of this chapter.*

Step 3: *Conclude with the estimates*

$$\hat{\overline{I}}_{\mathrm{a}}(\mathcal{X};\mathcal{O}) = -\frac{1}{n}\log_2 r_{\mathrm{a}}(o^n) - h(\mathcal{O}|\mathcal{X}) \tag{3.144}$$

and

$$\hat{\underline{I}}_{\mathrm{a}}(\mathcal{X};\mathcal{O}) = -\frac{1}{n}\log_2 r_{\mathrm{a}}(o^n) + \frac{1}{n}\log_2 w_{\mathrm{a}}(o^n|x_{1-m}^n). \tag{3.145}$$

Note that the term $h(\mathcal{O}|\mathcal{X})$ in the upper bound of (3.144) refers to the original channel and cannot be computed by means of the auxiliary-channel.

As the upper bound hinges on upper bounding $h(\mathcal{O})$. It can be tightened by maximizing the probability that the observed sequence (of the original channel) is generated by the auxiliary FSM, i.e.

$$\max_{\boldsymbol{\theta}(L)} r_{\mathrm{a}}\left(o^n;\boldsymbol{\theta}(L)\right). \tag{3.146}$$

This maximization or (unsupervised) learning of the auxiliary FSM is accomplished by the Baum-Welch training algorithm [62]. If the input sequence x^n is available, we can maximize the conditional probability, i.e.

$$\max_{\boldsymbol{\theta}(L)} r_{\mathrm{a}}\left(o^n|x^n;\boldsymbol{\theta}(L)\right), \tag{3.147}$$

and the reestimation formulas of the Baum-Welch algorithm amount to estimating the model parameters (supervised learning).

3.5.3 Reduced-State Version

Reduced-State Upper Bound

As the upper bound $\overline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ assumes that $h(\mathcal{O}|\mathcal{X})$ is known, a reduced-state upper bound on $I(\mathcal{X};\mathcal{O})$ is obtained by upper bounding $h(\mathcal{Y})$ as outlined in Theorem 3.2.

Reduced-State Lower Bound

If $h(\mathcal{O}|\mathcal{X})$ is known, a reduced-state lower bound on $I(\mathcal{X};\mathcal{O})$ is obtained simply by lower bounding $h(\mathcal{Y})$ as outlined in Theorem 3.3.

We can also obtain a reduced-state lower bound on $I(\mathcal{X};\mathcal{O})$, if $h(\mathcal{O}|\mathcal{X})$ is unknown. First, we note that the law of the auxiliary-channel $w_{\mathrm{a}}(\cdot|\cdot)$ in (3.142) can be any (possibly time-varying) FSM. Important is only that, the estimates of $h(\mathcal{O})$ and $h(\mathcal{O}|\mathcal{X})$ are both computed on the same FSM. If the FSM is time-varying, it may be not possible to invoke ergodic arguments. In this case, we simply obtain an estimate of (3.142) by stochastic averaging.

Chapter 4

Information Rates of Magnetic Recording Channel Models

The algorithms presented in the previous chapter will now be applied to various FSMs for the magnetic recording channel. Numerical results are provided along with interpretations. Unless otherwise stated, the "forward-only method" and a trellis length of $n = 10^6$ was used to compute estimates of the information rates.

4.1 Channel Models without Medium Noise

Partial-response class-4 (PR4) polynomials with AWGN at the output are widely used as model for the magnetic recording channel in today's hard-disk drives. PR4 target polynomials are of the form

$$g(D) = c \cdot (1 - D^2) \cdot (1 + D)^{\varepsilon}, \tag{4.1}$$

where $\varepsilon \in \{0, 1, 2, \ldots\}$ is the extension degree. The constant c normalizes the channel transfer function to 1.

Example 4.1 (Partial-Response Polynomials)
For $\varepsilon = 0$ we obtain the PR4 polynomial, for $\varepsilon = 1$ the expended PR4 (EPR4) polynomial, for $\varepsilon = 2$ the E^2PR4 polynomial, and so on. For $\varepsilon \in \{0, 1, 2\}$ the normalized polynomials are listed in Table 4.1. Hence, PR4 polynomials are FIR filters with a $(1 - D^2)$ factor.

4.1.1 FIR Channel Models

i.u.d. Information Rates

First, we look at the information rate under the assumption that the bipolar input is i.u.d., i.e. $Pr(X_k = -1) = Pr(X_k = +1) = 0.5$. The input process is therefore a 0.5-Bernoulli process, and the resulting information rate is termed *i.u.d. information rate*, $I_0(\mathcal{X};\mathcal{Y})$. The polynomials of selected channel models are listed in Table 4.1. The channel tap coefficients $g_0, g_1, \ldots, g_m$ are normalized such that $\sum_{k=0}^{m} g_k^2 = 1$. The channel CH6 in Tab. 4.1 was taken from Hirt's thesis [41] (where it is actually called "channel 4").

Channel name	Normalized impulse response
DICODE	$g(D) = (1 - D)/\sqrt{2}$
EPR4	$g(D) = (1 + D - D^2 - D^3)/2$
E^{2}PR4	$g(D) = (1 + 2D - 2D^3 - D^4)/\sqrt{10}$
CH6	$g(D) = 0.19 + 0.35D + 0.46D^2 + 0.5D^3 + 0.46D^4 + 0.35D^5 + 0.19D^6$

Table 4.1: Impulse responses of selected FIR channel models.

Fig. 4.1 shows the i.u.d. information rate of the FIR channel models from Tab. 4.1. The channel input energy is always $E_S = 1$, and the noise is AWGN with variance $N_0/2$. Note that, for the memoryless channel ($g(D) = 1$), whose i.u.d. information rate is also shown in Fig. 4.1, the i.u.d. information rate coincides with the BPSK channel capacity given by (2.110). The following observations can be made:

- For a given SNR=E_S/N_0, the i.u.d. information rate decreases with increasing memory, especially at high SNR where the low noise floor reveals the different channel models.
- At a rate of 0.9 bits/symbol, the i.u.d. information rate of the DICODE channel is 0.8 dB away from the capacity of the memoryless channel (BPSK capacity). The i.u.d. information rate of the E^{2}PR4 channel is another dB off.

Convergence Behavior

The estimates of the i.u.d. information rates in Fig. 4.1 were all computed by the "forward-only" method presented in the previous chapter. The convergence behavior of this method is illustrated in Fig. 4.2. Estimates of the i.u.d. information rate of the DICODE channel at 0 dB

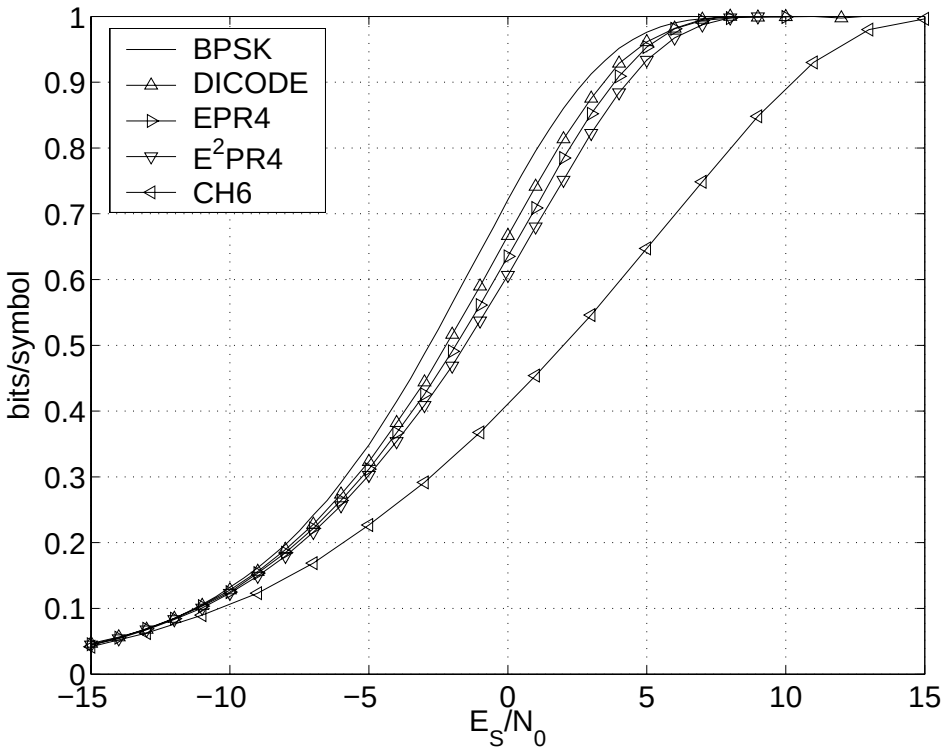

Figure 4.1: Estimated i.u.d. information rates of selected FIR channel models.

were computed 10 times, 100 times, and 1000 times; each time with a different random seed. The random seeds were selected in a nested way, such that the collection of the 100 estimates contains also the collection of the 10 estimates. Likewise, the collection of the 1000 estimates contains the collection of the 100 estimates as well as the collection of the 10 estimates. For various trellis lengths n, Fig. 4.2 shows the minimum and maximum estimate (denoted as "LB" and "UB") of the i.u.d. information rate among the collection with 10 estimates, the collection with 100 estimates, and the collection with 1000 estimates. We observe that the envelope from above/below decays/increases with $O(n^{-\frac{1}{2}})$ as predicted by theory.

Lower Bounds on the FSC Capacity

We now go beyond i.u.d. input. We compute the FSM capacity of the DICODE channel for the first extension of the Bernoulli input process, C_1^{FSM}, a Markov process with two states, and for the second extension, C_2^{FSM}, a Markov process with four states. We assume here that the iterative information rate maximization method presented in Chapter 3 converges to the FSM capacity after a sufficiently large number of iterations (see also the next subsection on this topic). Unless otherwise stated, we use the unconditioned version of the a-posteriori state-transition weights for the maximization procedure.

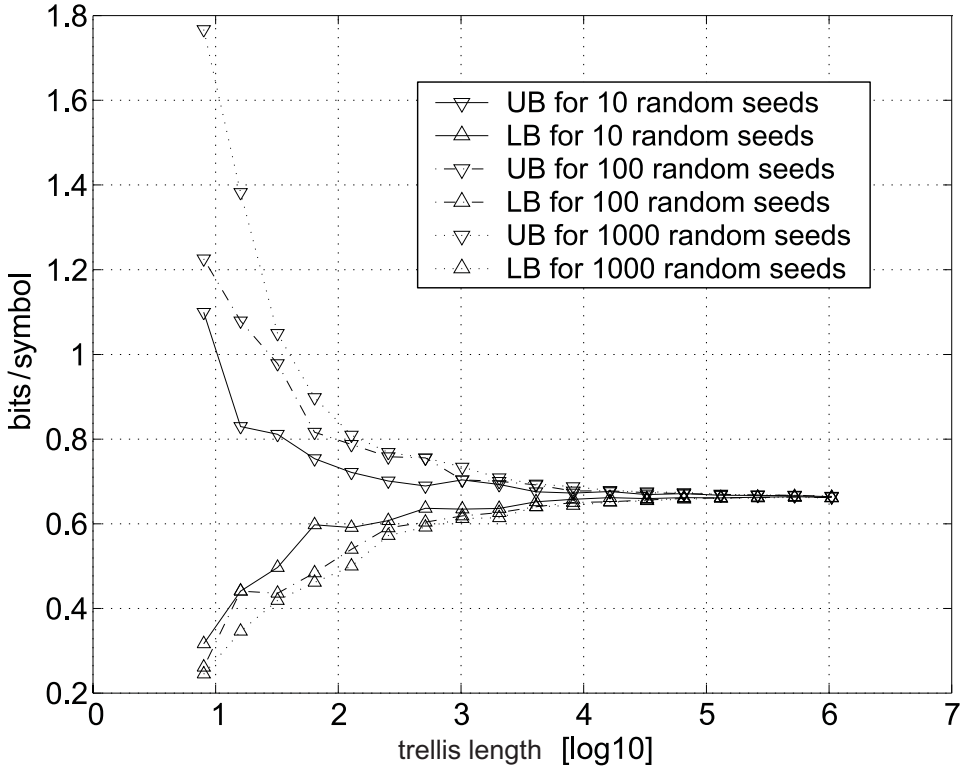

Figure 4.2: Convergence behavior of the forward-only method for the DICODE channel at 0 dB.

The results are shown in Fig. 4.3. It is easy to see that the i.u.d. input process is optimal for SNR $\to \infty$. At low SNR, however, non-uniform corrolated input processes produce more output symbols with high energy.

Also shown in Fig. 4.3 are the capacities of both the memoryless AWGN channel and the DICODE channel for nonbinary power-limited (Gaussian) input (denoted by "Gauss max"). The plot shows that, the optimized Markov processes achieve noticeably higher rates than the i.u.d. process does. These rates exceed even the capacity of the memoryless Gaussian channel in the low SNR regime. This is due to the fact that the channel impulse response is normalized to one. The normalization is quite arbitrary, but it guarantees that the channel is energy-preserving, i.e. the output energy equals the input energy for i.u.d. input. Some frequencies of the channel amplify, others attenuate the input spectrum compared with a flat (white) spectrum. By transmitting in the frequencies where the input is amplified, a noticeably higher SNR at the channel output is achieved. Note that this does not mean that the channel is an *active* element.

The eight state-transition probabilities associated with the memory-two input process are shown in Fig. 4.4 after 30 maximization steps of the iterative maximization method from Chapter 3 vs. E_S/N_0. We note that at high SNR uniformly distributed STPs are optimal

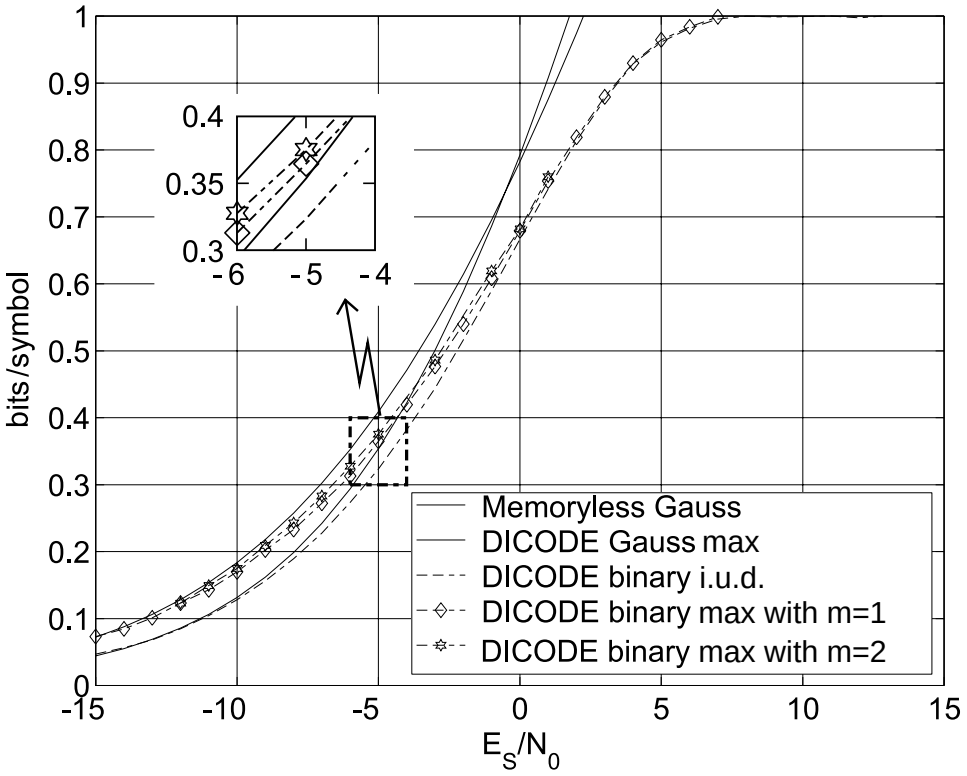

Figure 4.3: DICODE channel: Estimated information rates for various input processes.

whereas at low SNR some STPs are clearly preferred to others. The output symbols associated with the preferred STPs lead to a higher SNR at the channel output. Moreover, we note that at very low SNR some STPs converge nearly to one. The numbers along the STP axis in Fig. 4.4 correspond to the following state-transitions

$$1: \quad \mathbf{Q}(1,1) \qquad 5: \quad \mathbf{Q}(3,4) \tag{4.2}$$

$$2: \quad \mathbf{Q}(1,2) \qquad 6: \quad \mathbf{Q}(3,3) \tag{4.3}$$

$$3: \quad \mathbf{Q}(2,4) \qquad 7: \quad \mathbf{Q}(4,1) \tag{4.4}$$

$$4: \quad \mathbf{Q}(2,3) \qquad 8: \quad \mathbf{Q}(4,2). \tag{4.5}$$

These state-transitions are depicted in Fig. 4.5 with their associated input/output pair. The most probable state-transitions are depicted with solid lines, less probable ones with dashed, the even less ones with dot-dashed lines, and the least probable ones with dotted lines.

Similar results were obtained for the EPR4 and CH6 channels by systematically optimizing the state-transition probabilities. They are shown in Fig. 4.6 and Fig. 4.7 respectively. At rate one half, a memory-six input process improves the lower bound on the capacity of the CH6 channel by 2 dB (see Fig. 4.7).

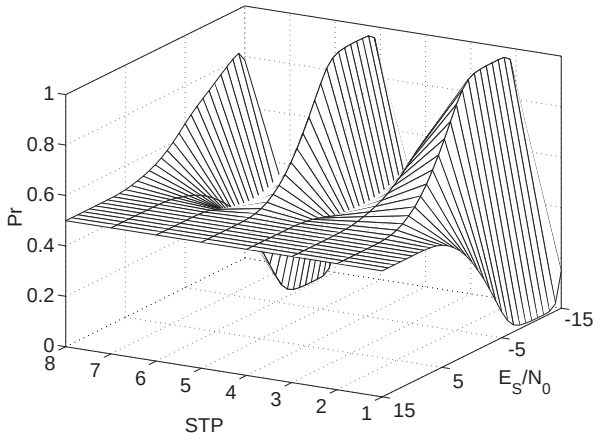

Figure 4.4: DICODE channel: Optimized STPs vs. E_S/N_0 after 30 iterations.

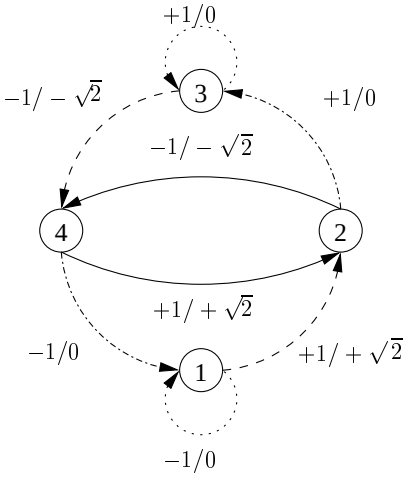

Figure 4.5: State-transition diagram for the memory-two Markov source.

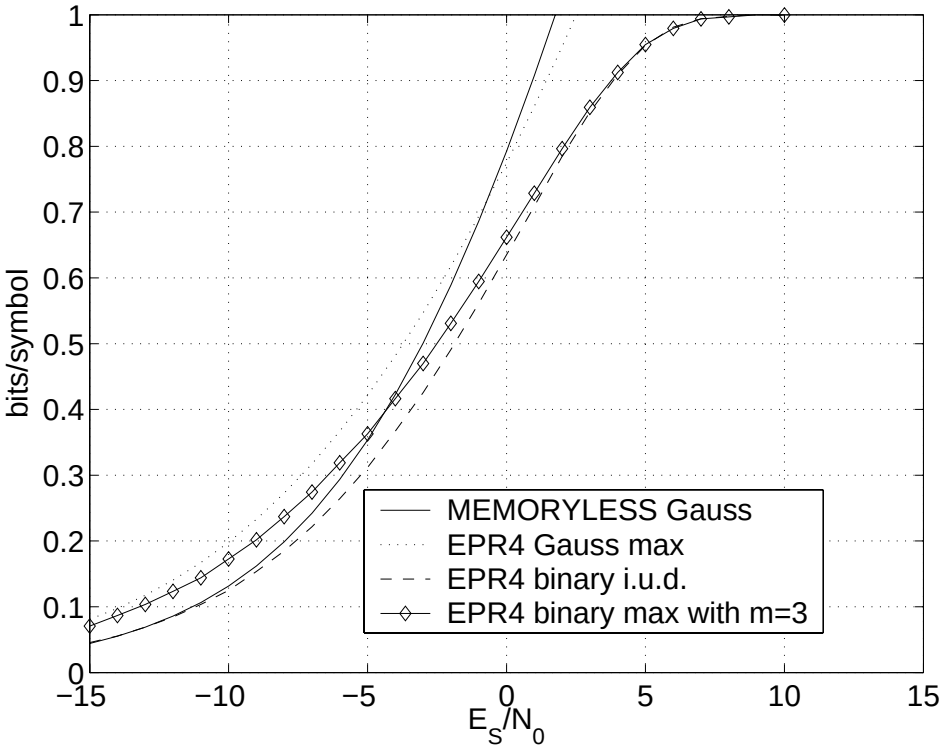

Figure 4.6: EPR4 channel: Estimated information rates for various input processes.

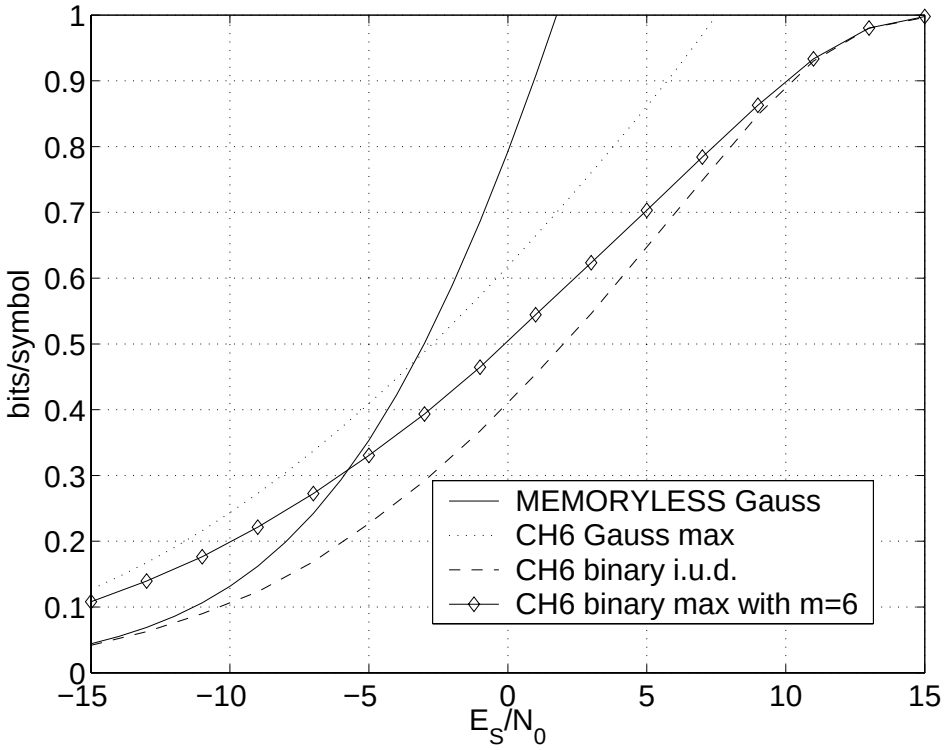

Figure 4.7: CH6 channel: Estimated information rates for various input processes.

Upper Bounds on the FSC Capacity

Upper bounds were computed for the DICODE channel and for a channel with impulse response $g_{\text{obs}}(D) = (1 - D/2)/\sqrt{5/4}$. This channel is termed *observable* because, given a finite output sequence, the input can be inferred in the absence of noise without ambiguities. This is not the case for the DICODE channel, because its trellis is quasi-catastrophic. Hence, the input of the DICODE channel can only be inferred from the entire output sequence including the starting state.

Upper bounds on the capacity of the DICODE and the observable channel with an eight-state input process, $\eta = 3$, and $M = 6$ are shown in Fig. 4.9 and Fig. 4.8, respectively. The "water-filling" upper bound, denoted in the figure as "Gauss max", is plotted as well. The lower bounds, C_3^{FSM}, were computed with the iterative information rate maximzation method.

The upper bound implies estimating the state sequence based on a output sequence of finite length. The tightness of the upper bound depends therefore on how well this estimation can be done. For high SNR, the upper bound is not tight for the DICODE channel. The overshoot is proportional to $1/(M + 1)$.

In his thesis [59], Pfister conjectured a tight upper bound using a slightly different approach. Numerical results for the DICODE channel are excellent [59].

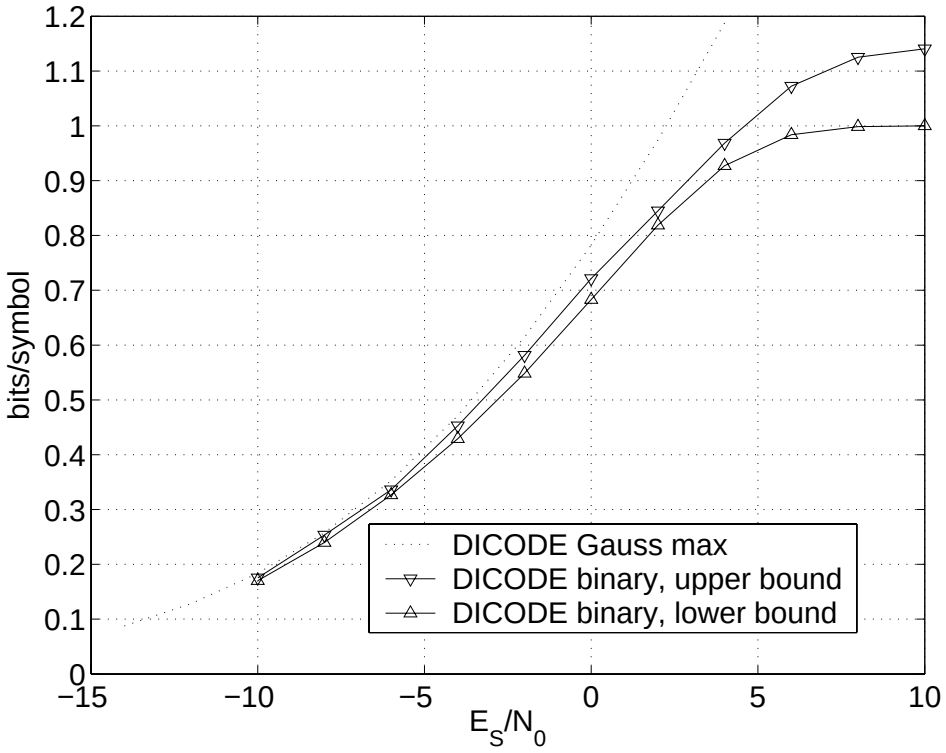

Figure 4.8: Estimated upper and lower bound on C^{FSC} of the DICODE channel.

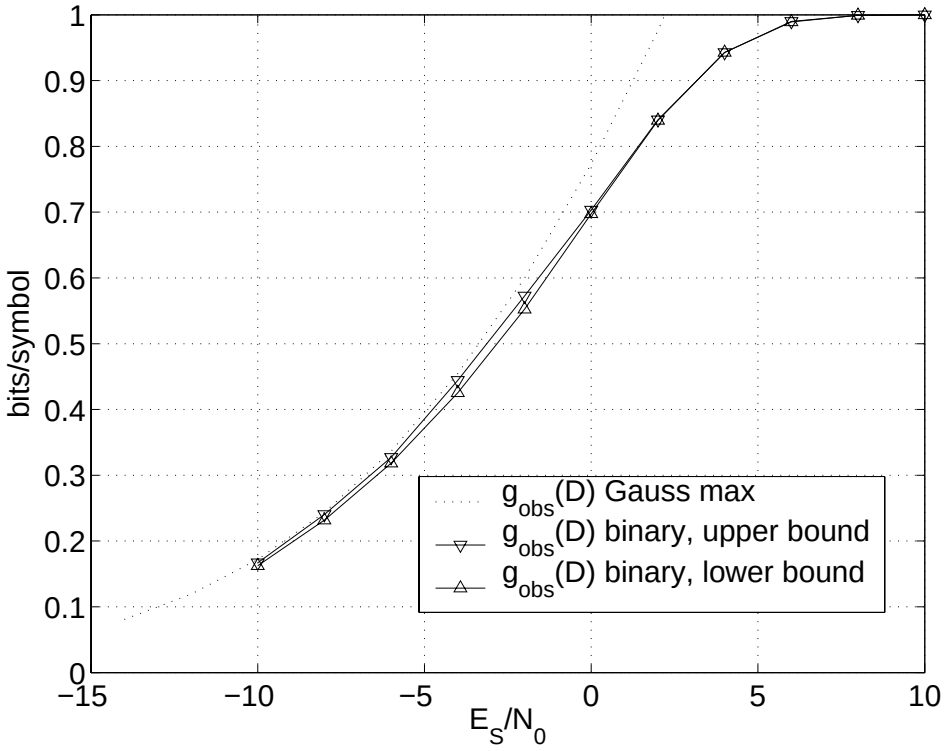

Figure 4.9: Estimated upper and lower bound on C^{FSC} of the observable channel.

4.1.2 Maximization of the FSM Information Rate

"Memory Increases Capacity"

This famous quote is attributed to Wolfowitz [77] and can be verified for power-limited input signals by means of the "water-filling" technique. Numerical results are given for instance in [41]. Therein, it is shown that at low SNR an increase in memory leads to a substantial increase in capacity. At high SNR, it is the opposite, i.e. an increase in memory leads to a substantial decrease in capacity. Thus, the capacity of a channel with memory lies at low SNR above the capacity of a memoryless channel, approaches it from above with increasing SNR, crosses it at some SNR point, and from thereon lies strictely below the capacity of the memoryless channel (see also "Gauss-max"-curves in Figs. 4.3, 4.6, and 4.7). This is a consequence of the normalization of the channel transfer function.

We investigate mainly the behavior of the iterative information rate maximization method from Kavčić. Fig. 4.10 shows the increase in information rate for the DICODE channel with a memory-one (DIM1) and a memory-two (DIM2) Markov input process at –10 dB versus the number of iterations. For a given set of STPs, the information rate was computed with two different methods: using the forward-only sum-product algorithm (SPA) and evaluating the largest eigenvalue (EW) of the noisy adjacency matrix according to (3.95). Note that the EW method is only a valid information rate at stationary points of the iterative information rate maximization method.

We make the following observations:

- For both input processes, the iterative procedure reaches a saturation point and remains there.
- The memory-two Markov input process leads to a noticeably higher information rate than the memory-one Markov input process.
- At the beginning (iterations 1 to 6), the information rate evaluated with the EW method is higher than the one evaluated with the SPA. It lies roughly halfway to the information rate of the next iteration evaluated with the SPA. Moreover, it saturates at the same level as the information rate evaluated with the SPA.

Iterative Behavior

In Fig. 4.11, the differences between various information rate computation methods are plotted versus the number of iterations of the interative information rate maximization method for

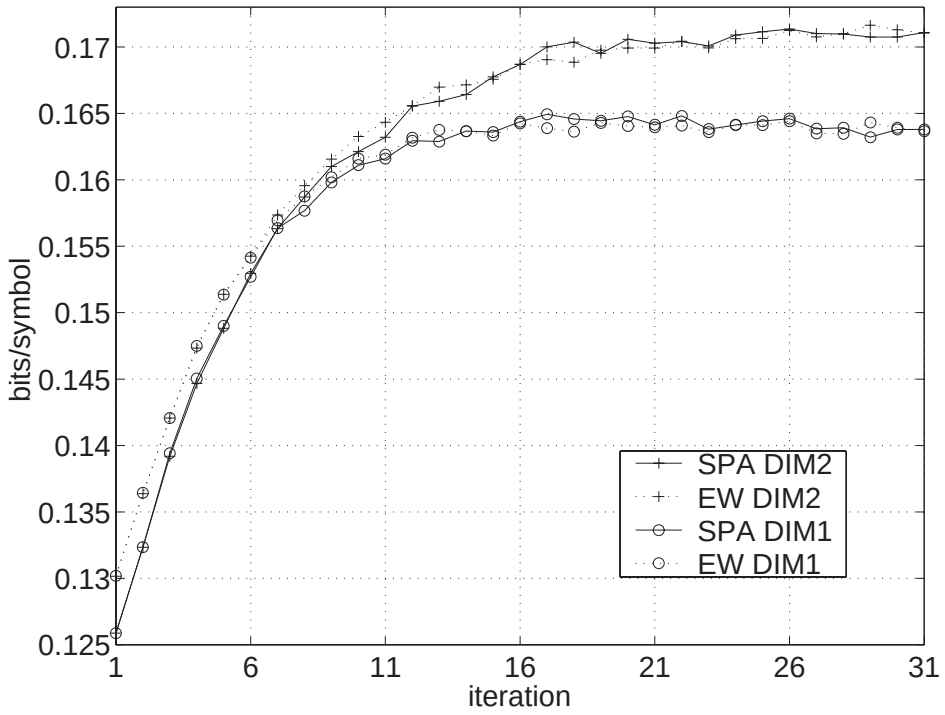

Figure 4.10: Estimated information rates vs. number of iterations of the DICODE channel at –10 dB. At iteration "1", the i.u.d. information rates are plotted. At iteration "31", information rates are plotted that were obtained after applying 30 steps of the iterative information rate maximization method.

the DICODE channel with a memory-two Markov input process at -10 dB. The reference method is in all cases the forward-only sum-product algorithm (SPA). The other methods are:

- "EW": the information rate is computed via the largest eigenvalue of the noisy adjacency matrix according to (3.95).
- "F EW 1": the information rate is computed using (3.82).
- "F EW 2": the information rate is computed via (3.82), but with "new" (updated) state probabilities and "old" (not updated) STPs. The state probabilities are obtained via new (updated) STPs given by (3.96).
- "F EW 3": the information rate is computed using (3.82), but with "new" (updated) state and STPs. This method is mathematically equivalent to the "EW" method.

The methods "EW", "F EW 2", and "F EW 3" deliver only a meaningful information rate if a local maximum (stationary point) is achieved; otherwise, the a-posteriori state-transition weights are generated from a different source.

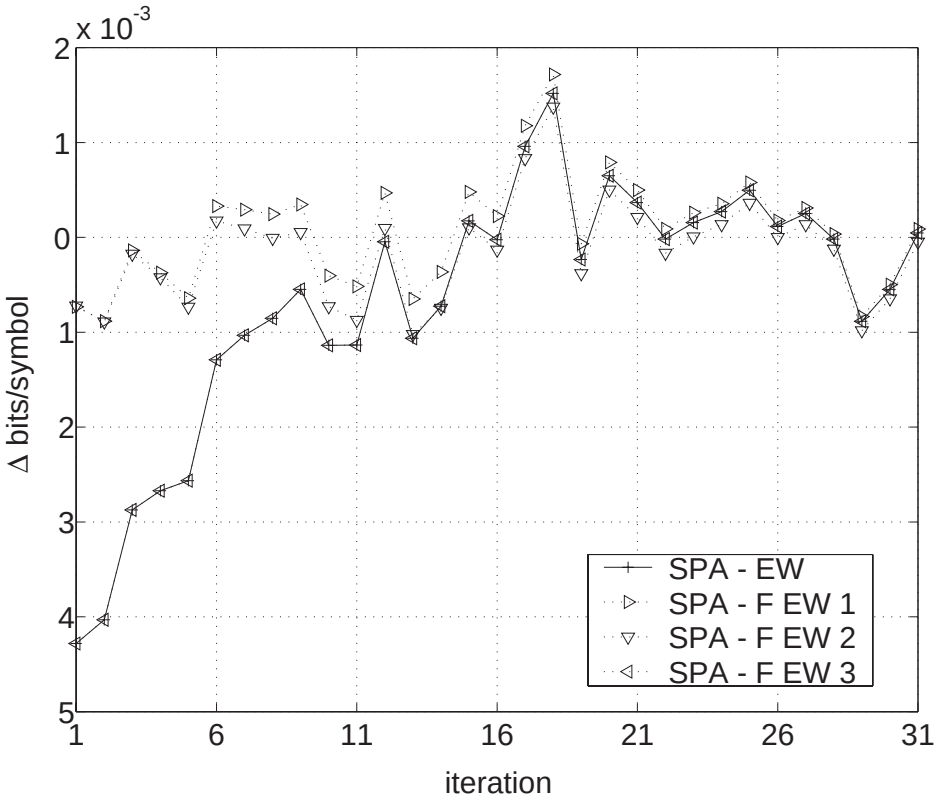

Figure 4.11: Differences in bits/symbol for various information rate computation methods vs. number of iterations for the DICODE channel with a memory-two Markov input process at -10 dB. Iteration "1" means i.u.d. input. Iteration "2" means one update of the state-transition probabilities with the iterative information rate maximization method.

We observe the following:

- The differences are quite small.
- With increasing number of iterations, all differences become the same and their magnitudes decrease. Thus, all methods reach the same saturation point.
- The difference between the "EW" and the "SPA" method diminishes during the maximization process. This strongly suggests that the iterative information maximization method indeed finds a local maximum.
- Method "EW" and "F EW3" lead to the same numerical results (as it should be).
- The difference between "SPA" and "F EW 1" originates from the fact that the $\mathbf{A}(i,j)$s in (3.82) are computed with a forward and a backward run of the sum-product algorithm.
- The difference between "SPA" and "F EW 1" on the one hand and "SPA" and "F EW 2" on the other hand is almost the same. For the "F EW 2" method, new (updated) state probabilities are used that result from the maximized state-transition probabilities of the noisy adjacency matrix. Hence, the state probabilities fit the noisy adjacency matrix, but the improvement is negligible (at least for the DICODE channel with a memory-two Markov input process). The real improvement comes from utilizing the maximized STPs.

A $(0,2)$-RLL Source Observed through AWGN

We consider a $(0,2)$-RLL source given by the adjacency matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \tag{4.6}$$

observed through AWGN. The trellis of the $(0,2)$-RLL is asymmetric, i.e. $\mathbf{A} \neq \mathbf{A}^T$. The hope is that if the iterative information rate maximization method (denoted by IIRMM in Fig. 4.12 and Fig. 4.13) works on this trellis, it will work on any other trellis as well. The STPs of the $(0,2)$-RLL source were optimized with the IIRMM method and the gradient-based method outlined in Chapter 3 for an SNR of +10 dB.

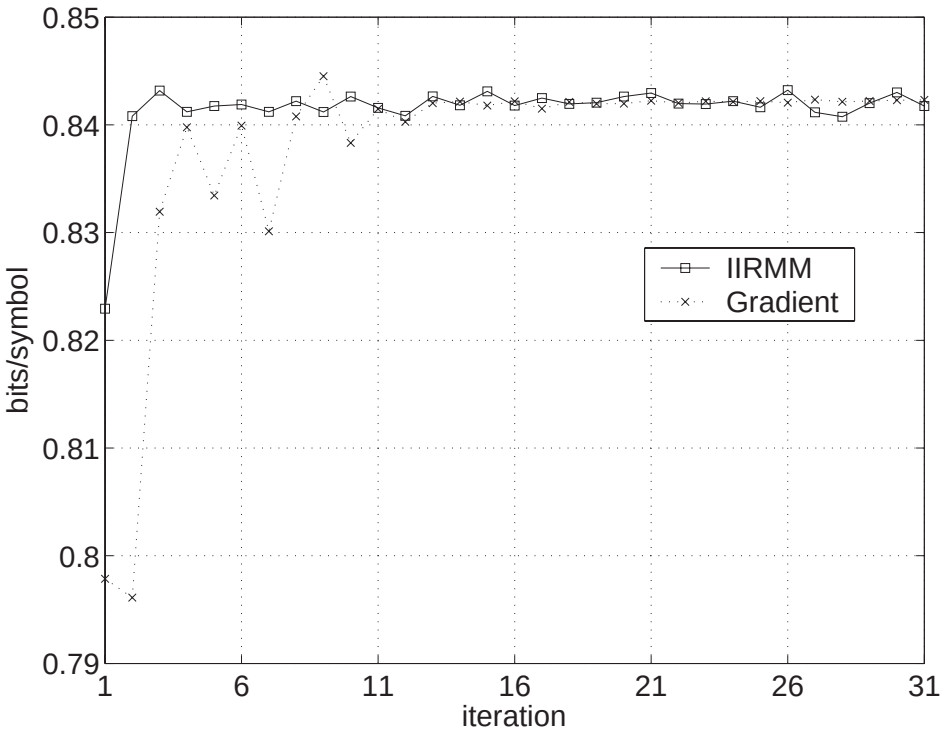

Figure 4.12: Estimated information rates of a $(0, 2)$-RLL source observed through AWGN: IIRMM and gradient-based method.

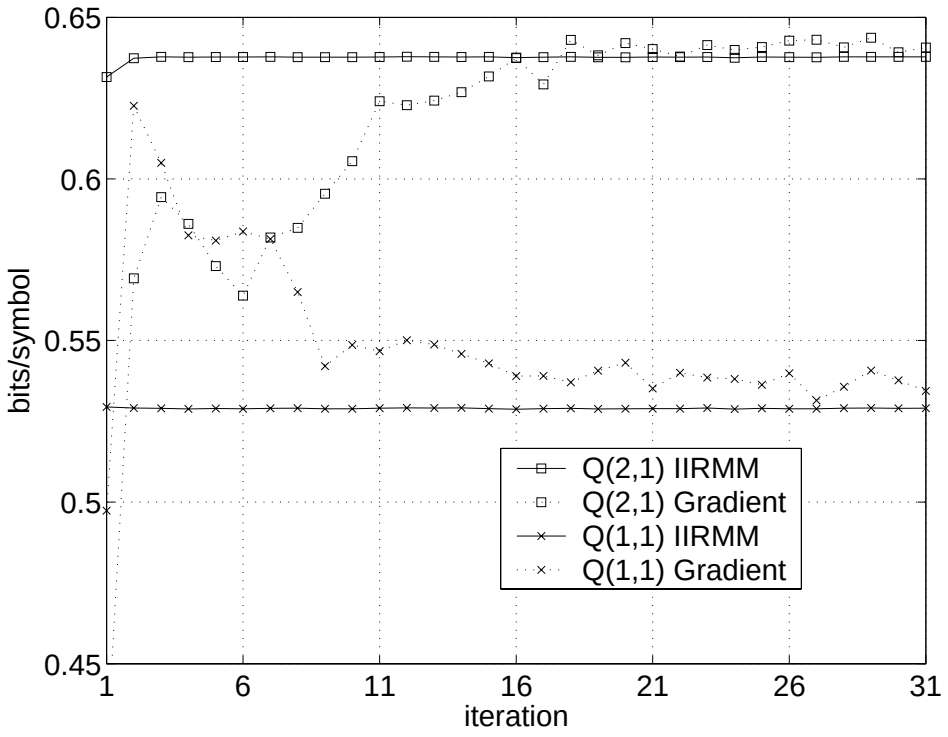

Figure 4.13: STPs of a $(0, 2)$-RLL source observed through AWGN: IIRMM and gradient-based method.

The estimated information rates are plotted vs. the number of iterations in Fig. 4.12. The corresponding STPs are depicted in Fig. 4.13. We make the following observations:

- Because of the noise, both methods lead to information rates below the capacity of the noiseless $(0, 2)$-RLL source, which is 0.8791 bits/symbol.

- Both iterative methods converge to exactly the same information rate. The numerical values of the gradient-based method fluctuate strongly at the beginning of the maximization procedure, whereas the iterative maximization method delivers reliable results right from the beginning. The trellis length was kept constant for the iterative method at $n = 10^6$ and the trellis length of the gradient-based method was steadily increased from $n = 10^4$ to $n = 10^9$.

- The STPs of the gradient-based method converge slowly (compared to the information rate in Fig. 4.12) to the ones obtained with the iterative information rate maximization method. This indicates that the information rate is not terribly sensitive to varying STPs.

- The STPs of the gradient-based method do not converge exactly to the ones obtained from the iterative method, because of the finite resolution $\mathbf{\Delta}(2, 1)$ and $\mathbf{\Delta}(1, 1)$ respectively.

4.1.3 Information Rates of General Channels

We provide some examples for the method presented in the last section of Chapter 3. The observed data are collected from channel simulations performed on a computer assuming an i.u.d. input process.

EPR4 Model as Original Channel

Let the original channel be a linear EPR4 model with AWGN at the output. Estimates of $\overline{I}_{\mathrm{a}}(\mathcal{X}; \mathcal{O})$ and $\underline{I}_{\mathrm{a}}(\mathcal{X}; \mathcal{O})$ are easily computed; they are shown in Fig. 4.14 vs. the memory m of the auxiliary FSM for two different SNR values. We note that from $m = 3$ on, $\hat{\overline{I}}_{\mathrm{a}}(\mathcal{X}; \mathcal{O})$ and $\hat{\underline{I}}_{\mathrm{a}}(\mathcal{X}; \mathcal{O})$ coincide — the original channel has memory three — and equal the i.u.d. information rate for the EPR4 channel at these SNRs (see Fig. 4.6).

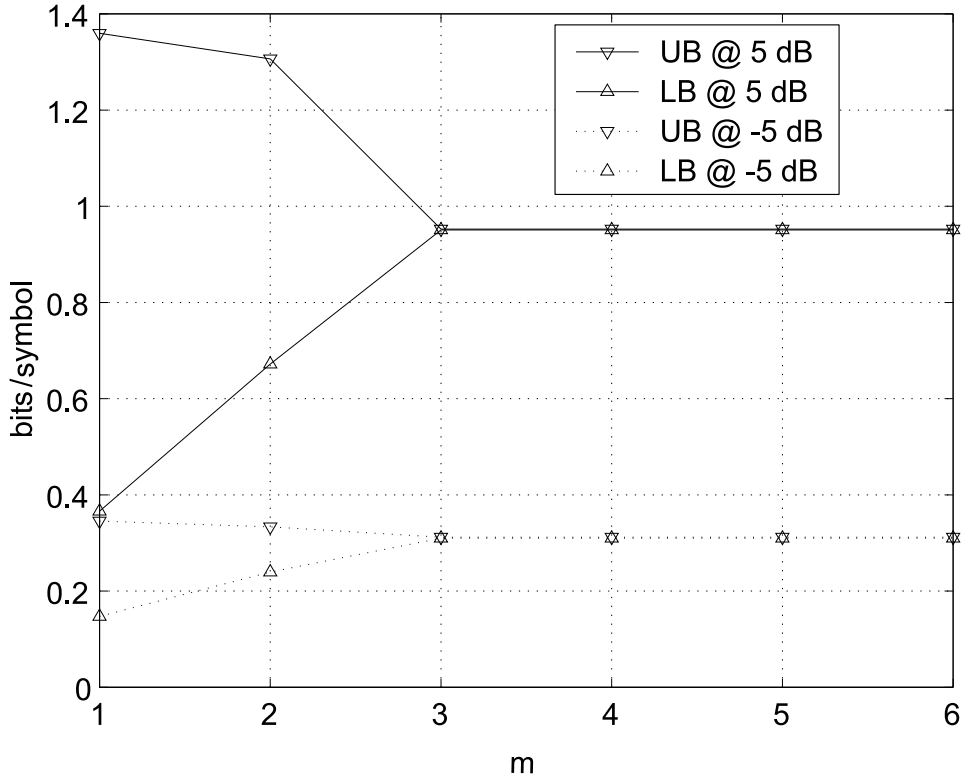

Figure 4.14: Estimates of $\overline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ and $\underline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ (denoted by UB and LB, respectively) for the EPR4 channel vs. m.

Linear Lorentzian Channel as Original Channel

Next, we assume that the original channel is a linear infinite-impulse response channel with AWGN at the output. The impulse response is the "Lorentzian" pulse, which is characterized by the parameter PW50. The ratio PW50/T, where T is the nominal duration of a data symbol, is a measure of the normalized linear density in a hard-disk system. For this example, we chose PW50/$T = 2.0$ and 3.2. We also normalized the channel transfer function to one.

In Fig. 4.15, estimates of $\overline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ and $\underline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ are plotted vs. the SNR for $m = 10$. At low SNR, the bounds are very tight, but at high SNR they become loose due to the model mismatch between the original channel and the FSM. This model mismatch is elucidated in Fig. 4.16, where estimates of these bounds are plotted vs. the memory m of the FSM. The asymptotic gap for $m \to \infty$ between $\overline{I}_{\mathrm{a}}(\mathcal{X}|\mathcal{O})$ and $\underline{I}_{\mathrm{a}}(\mathcal{X}|\mathcal{O})$ depends on the suitability of the class of FSMs chosen; if the original channel *is* such an FSM, as is the case of the EPR4 channel, the gap vanishes. We note that less memory is required at low SNRs, where the model mismatch between the original channel and the FSM is concealed by the noise. Moreover, for a given SNR less memory is required to obtain tight bounds for the narrower Lorentzian pulse with PW50/$T = 2.0$.

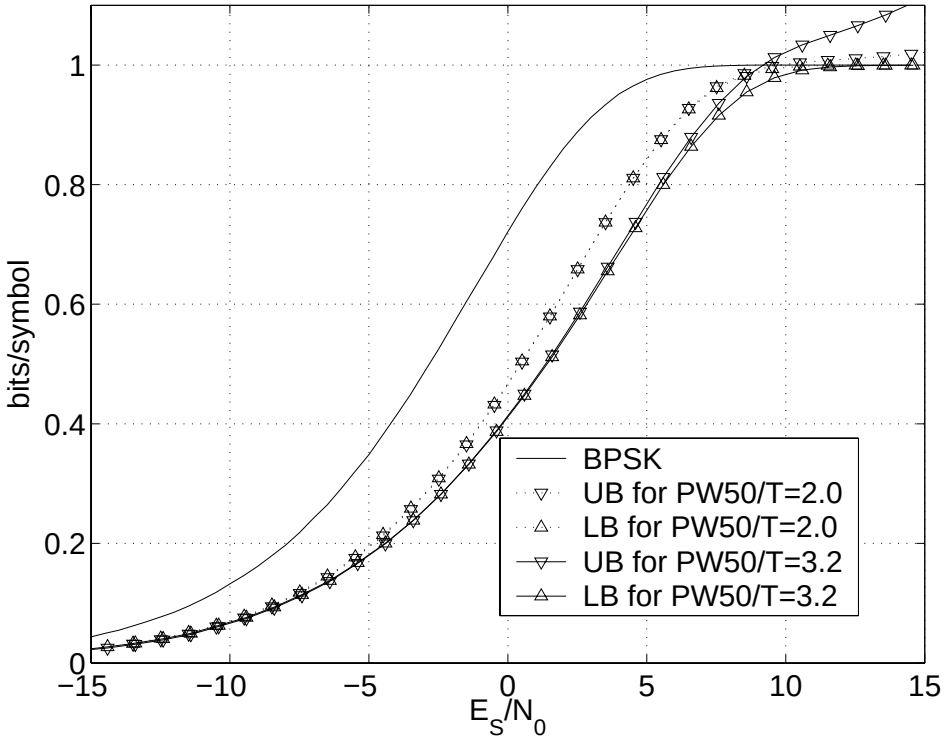

Figure 4.15: Estimates of $\overline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ and $\underline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ (denoted by UB and LB, respectively) for the normalized linear Lorentzian channel. The memory of the auxiliary FSM is $m = 10$. Also shown is the capacity of BPSK.

1st Order Autoregressive Channel as Original Channel

Now, we consider linear intersymbol interference channels of the type

$$Y_k = \sum_{i=0}^{M} g_i X_{k-i} + Z_k \tag{4.7}$$

with large memory $M \in \mathbb{Z} \cup \{\infty\}$, with fixed real channels coefficients, i.e. $g_0, g_1, \ldots, g_M \in \mathbb{R}$, and where $\mathcal{Z} = (Z_1, Z_2, \ldots)$ is a white Gaussian noise process with variance σ^2. The channel input process $\mathcal{X} = (X_1, X_2, \ldots)$ will be a sequence of i.u.d. random variables taking on values in $\{+1, -1\}$. The channel state at time k is the vector $(X_{k-1}, X_{k-2}, \ldots, X_{k-M})$ of the past M channel inputs. Each such state is thus a binary M-tuple. We will denote such M-tuples/states by $s_k = (x_{k-1}, x_{k-2}, \ldots, x_{k-M})$ with $x_\ell \in \{+1, -1\}$ for $1 \leq \ell \leq M$.

As an example of such a channel, we will use a first order autoregressive channel with impulse response

$$g(D) = \frac{1}{(1 - \alpha D)} = 1 + \alpha D + \alpha^2 D^2 + \ldots \tag{4.8}$$

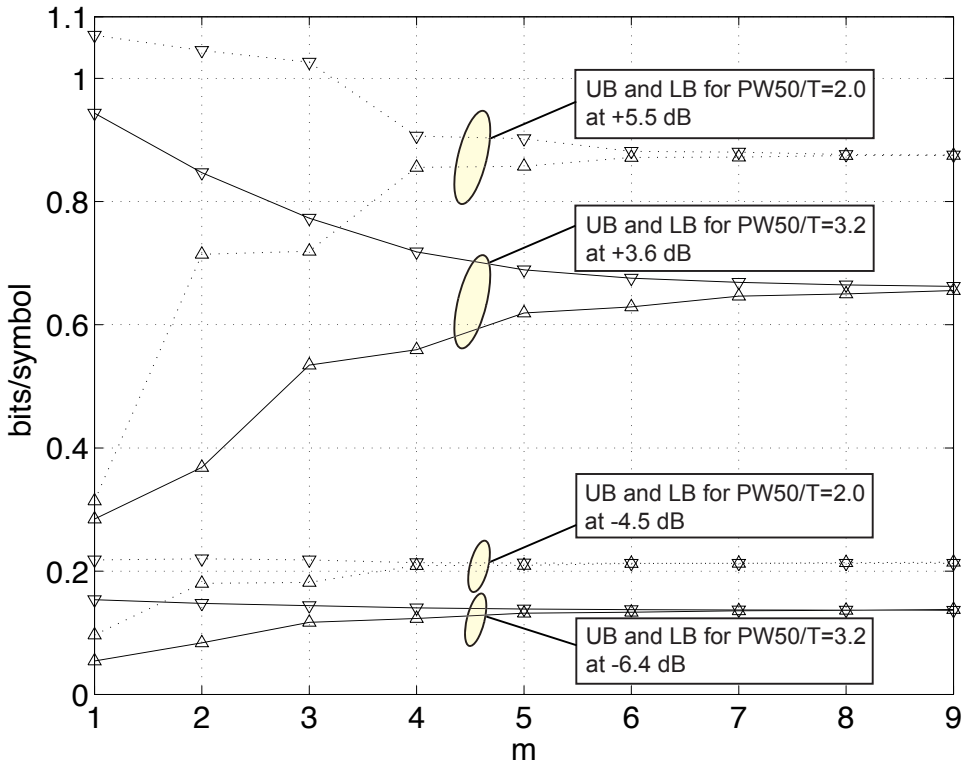

Figure 4.16: Estimated upper and lower bounds on the i.u.d. information rate for the normalized linear Lorentzian channel vs. the memory m. The upper four curves experience the same amount of AWGN. The same holds for the lower four curves. Due to channel normalization the SNR values for PW50/$T = 2.0$ and PW50/$T = 3.2$ differ (see also Fig. 4.15).

with $0 \leq \alpha < 1$ as illustrated in Fig. 4.17. A natural finite-state approximation is obtained by truncating the impulse response. We call this approximation "linear shift register" (LSR). Another finite-state approximation is obtained by inserting a quantizer in the feedback loop as shown in Fig. 4.18. We call this approximation "quantizer" (Q). Note that the channel in Fig. 4.18 is nonlinear.

Some numerical results for this example are shown in Fig. 4.19 for $\alpha = 0.8$ and $\sigma^2 = 1$. The figure shows estimates of $\overline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ and $\underline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$, both for the truncated-impulse response model and for the quantized-feedback model. The horizontal axis shows the memory m of the FSM. The quantizer in Fig. 4.18 was chosen to be a uniform quantizer optimized to give as good bounds as possible, i.e. the uniform distance between the quantizer levels was optimized. The noise samples $\tilde{Z}_k$ of the FSM in Fig. 4.18 were obtained by training the FSM. As Fig. 4.19 shows, the quantized-feedback model yields better bounds with less states than the truncated-impulse response model.

In Fig. 4.20, estimates of $\overline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ and $\underline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ are shown vs. the SNR. The bounds are

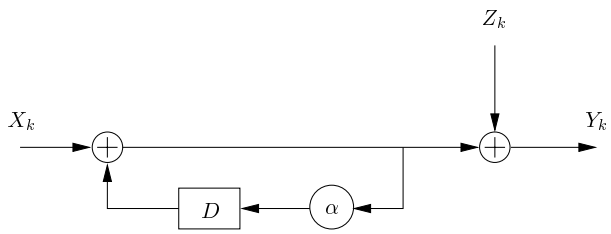

Figure 4.17: A simple non-finite-state channel.

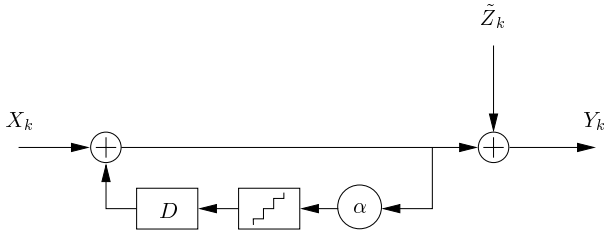

Figure 4.18: A quantized version of the channel in Figure 4.17.

indistinguishable! Fig. 4.20 also demonstrates that it is "easier" to find tight upper and lower bounds for the exponentially decaying infinite-impulse response of the 1st order autoregressive model than for the infinite-impulse response of the Lorentzian model that decays like a rational function. The gap caused by the model mismatch at high SNR disappers for the 1st order autoregressive model completely (compare also with Fig. 4.15).

A Reduced-State Upper Bound on the Information Rate of the Original Channel
The noiseless time-k output of the 1st order autoregressive channel is

$$-\frac{1}{1-\alpha} \leq y_k \leq \frac{1}{1-\alpha}. \tag{4.9}$$

For $\alpha \to 1$, the uniform quantizer in the feedback loop (see Fig. 4.18) must consists of many quantization levels to cope with the large output range (otherwise the quantization loss is big). On the other hand, the output range changes moderately from time step to time step. Consequently, many states of the uniform quantizer have a state metric near zero and contribute only marginally to the entropy computation of $h(\mathcal{Y})$ (see Fig. 4.21). Thus, recalling Theorem 3.2, we obtain a reduced-state upper bound on $h(\mathcal{Y})$ and thus on $I(\mathcal{X};\mathcal{O})$ simply by omitting states with small metric.

Assume now that, at time zero, the 1st order autoregressive channel is in some fixed initial state. At time one, there will be two states; at time two, there will be four states, etc.. We track all these states with the forward sum-product algorithm until there are too many of them. From then on, we keep only a predetermined fixed number of the states with the largest metric (i.e. the "best" ones) and continue to expand and collect these states with largest metric. By expanding the most likely states, the quantizer levels are *adjusted* at each time step. Thus, the output range of the quantizer varies from time step to time step while sailing along states with the largest metric (see Fig. 4.21). This upper bound is termed "adaptive reduced-state upper bound", or ARSUB for short. Numerical results in Fig. 4.19 and Fig. 4.20 show that it is very efficient.

A Reduced-State Lower Bound on the Information Rate of the Original Channel Recalling Theorem 3.3, we obtain a lower bound on $I(\mathcal{X};\mathcal{O})$ by merging states. As an example, we will consider *merged* states of the form

$$\begin{aligned} s_{k-1} &= (x_{k-1},\ldots,x_{k-m}) \\ &\triangleq \bigcup_{x_{k-M}^{k-(m+1)}} \{(x_{k-1},\ldots,x_{k-m},x_{k-(m+1)},\ldots,x_{k-M})\} \end{aligned} \tag{4.10}$$

for some positive interger $m < M$. In the trellis, the successors of some merged state $s_{k-1} = (x_{k-1},\ldots,x_{k-m})$ are the two merged states $s_k = (+1, x_{k-1},\ldots,x_{k-(m-1)})$ and $s'_k = (-1, x_{k-1},\ldots,x_{k-(m-1)})$.

We begin by assuming that the channel is in some known state at time zero. At time one, there will be two states; at time two, there will be four states, etc.. We first track all of these states, until there are too many of them. From that moment on, we merge states into the form given by (4.10) and we keep expanding and merging merged states.

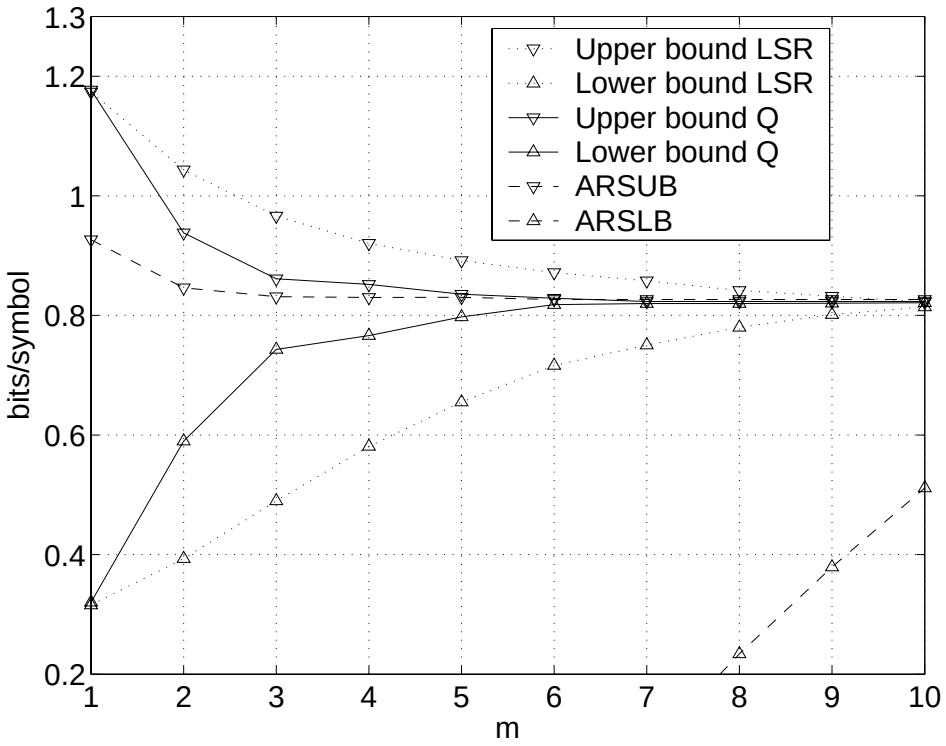

Figure 4.19: Estimated upper and lower bounds on the information rate of the 1st order autoregressive channel model vs. m.

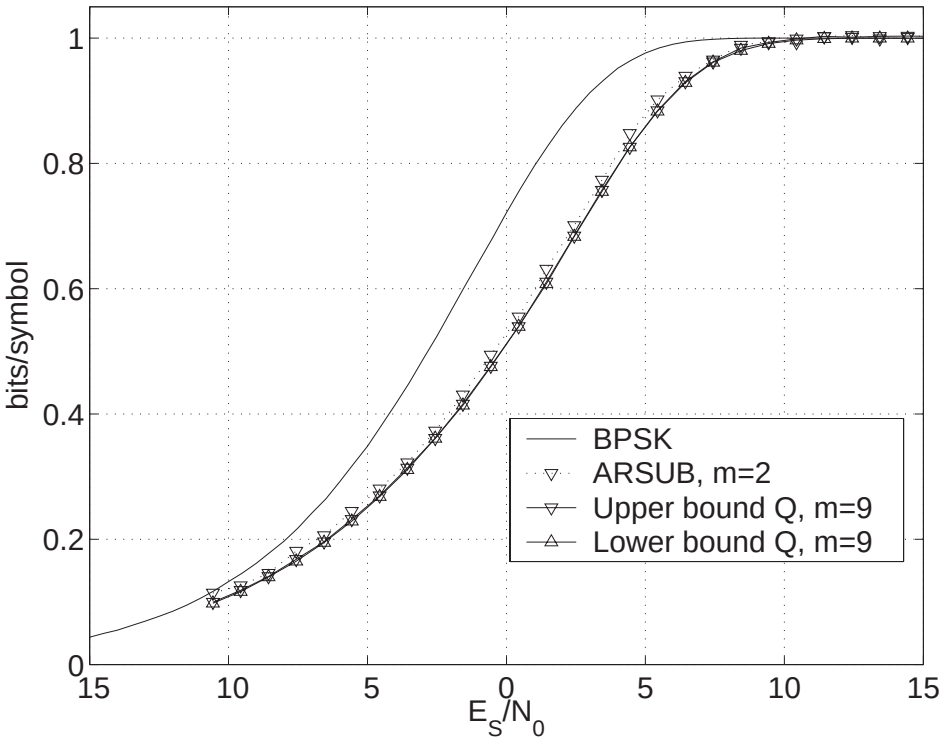

Figure 4.20: Estimates of $\overline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ and $\underline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$ for the 1st order autoregressive channel model vs. the SNR. The channel transfer function is normalized, i.e. $g(D) = \frac{1}{1-\alpha^2} \cdot \frac{1}{1-\alpha D}$.

The following merging strategy was used. First, a state metric is assigned to all states at time-k. Secondly, two states are called a *candidate pair* if they fulfill (4.10), i.e. if for both states $x_{k-(m+1)}, \ldots, x_{k-M}$ are identical. Thirdly, a candidate pair of states is merged according to (4.10) provided that the sum of their state metrics is the smallest among all possible candidate pairs. The set of states at time k is now reduced by one and a new list of candidate pairs is created. We continue merging until a desired number of remaining states at time k is achieved. The set of (unmerged) states at time $k+1$ is then given by the successors of the set of the remaining states at time k.

Assuming AWGN, we have for a non-merged state s_{k-1}

$$w(y_k|x_k, S_{k-1} = s_{k-1}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y_k-\tau)^2/(2\sigma^2)} \tag{4.11}$$

with τ

$$\tau \triangleq g_0 x_k + \sum_{\ell=1}^{M} g_\ell x_\ell. \tag{4.12}$$

For some merged state $s_{k-1} = (x_{k-1}, \ldots, x_{k-m})$, we replace τ by the interval $[\tau_{\mathrm{L}}, \tau_{\mathrm{U}}]$ with

$$\tau_{\mathrm{U}} \triangleq g_0 x_k + \sum_{\ell=1}^{m} g_\ell x_\ell + \sum_{\ell=m+1}^{M} |g_\ell| \tag{4.13}$$

and

$$\tau_{\mathrm{L}} \triangleq g_0 x_k + \sum_{\ell=1}^{m} g_\ell x_\ell - \sum_{\ell=m+1}^{M} |g_\ell|. \tag{4.14}$$

Eq. (4.11) is upper bounded by maximizing it over $\tau = [\tau_{\mathrm{L}}, \tau_{\mathrm{U}}]$. The maximizer $\hat{\tau}$ is then to

$$\hat{\tau} = \begin{cases} \tau_{\mathrm{L}} & \text{if} \quad y_k < \tau_{\mathrm{L}} \\ \tau_{\mathrm{U}} & \text{if} \quad y_k > \tau_{\mathrm{U}} \\ y_k & \text{else} \end{cases}. \tag{4.15}$$

Thus, (4.11) is maximized in particular if the channel symbols fall in the interval $[\tau_{\mathrm{L}}, \tau_{\mathrm{U}}]$. For our 1st order autoregressive channel, the range of this interval amounts to $2 \cdot \frac{\alpha^{m+1} - \alpha^{M+1}}{1-\alpha}$ with $M = \infty$. In contrary to the adaptive reduced-state upper bound, this interval does not change from time step to time step. Thus, we need a large quantizer to make this interval small and consequently to tighten the lower bound (for $\alpha \to 1$). This lower bound is termed "adaptive reduced-state lower bound", or ARSLB for short. Some numerical results are shown in Fig. 4.19.

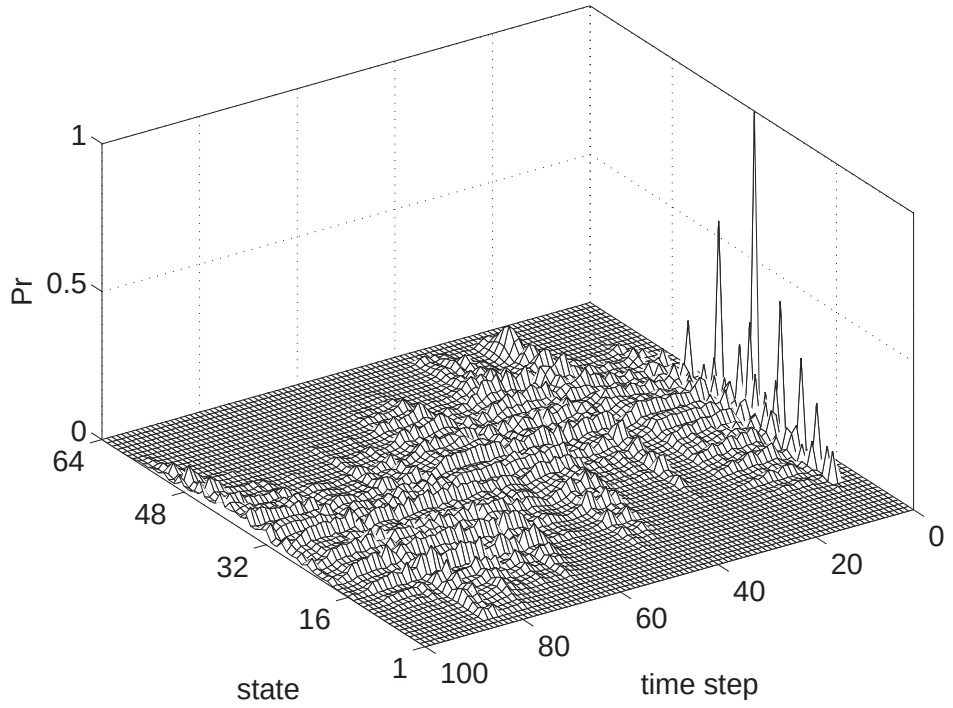

Figure 4.21: Forward state metrics from time step 0 up to time step 100 of the 1st order autoregressive channel using a 64-state quantizer.

4.2 Channel Models for Medium Noise

In this section, we investigate the effect of medium noise on the information rate. French and Wolf modeled medium noise as stationary correlated Gaussian noise [33] that is independent of the input data. Under these assumptions, they found that medium noise is preferable if it is shaped like the transmitted signal (Jensen-inequality).

However, medium noise is data-dependent, non-stationary, and not Gaussian distributed. Our approach is to train an FSM (representing a parameterized Gaussian mixture density with $\xi = 8$) with synthetically generated waveforms (of a microtrack model [20]) and then compute estimates of the lower bound $\underline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$, by means of the forward sum-product algorithm.

In [48, 49], an autoregressive-noise model was used to model medium noise. The autoregressive-noise model turns out to be a special case of our FSM. This can be seen from the following considerations. First, remember that the state at time-k of our FSM with memory m is build from the last m inputs, i.e. $s_k = (x_{k-1}, \ldots, x_{k-m})$. Let now $Y_k = X_k + Z'_k$ with channel input $X_k \in \{+1, -1\}$ and with $Z'_k = Z_k - \sum_{i=1}^{M} g_i Z'_{k-1}$, where $\mathcal{Z} = (Z_1, Z_2, \ldots)$ is white Gaussian noise and g_i are real fixed coefficients. In other words, $\mathcal{Z}' = (Z'_1, Z'_2, \ldots)$ is colored noise obtained by filtering the white noise process $\mathcal{Z}$ by an autoregressive filter with transfer function $1/g(D)$ with $g(D) = 1 + g_1 D + \ldots + g_M D^M$.
Let us assume that both $g(D)$ and $1/g(D)$ are stable. In this case, we realize from Fig. 4.22 that the original channel (left part) is equivalent to the channel in the right part of Fig. 4.22 and that the filter $1/g(D)$ in the latter channel can be dropped without affecting the information rate $I(\mathcal{X};\mathcal{Y})$ (data processing lemma).
Moreover, the joint FSM (denoted as FSM' in Fig. 4.22 right part) consisting for the original FSM and $g(D)$ is again an FSM. This can be seen from the fact that $g(D)$ is an FIR filter. Thus, the output V'_k is a function (via V_k) of the input X_k (right part in Fig. 4.22) like V_k is a function of the input X_k (left part in Fig. 4.22). Our FSM can therefore also model correlated

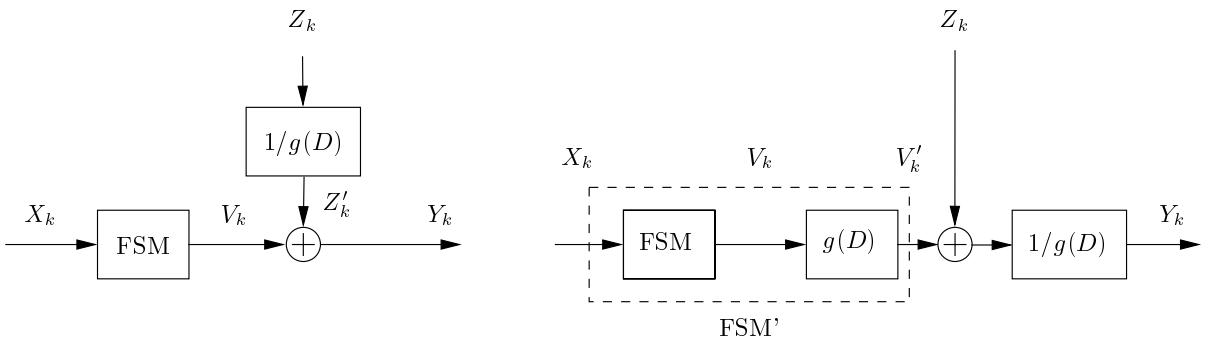

Figure 4.22: Autoregressive-noise channel (left) and equivalent channel (right).

Gaussian mixture noise.

After this discussion on the FSM, we detail a bit on the computer model (microtrack model) that was used to emulate waveforms from magnetic hard-disk drives.

4.2.1 The Microtrack Model

The microtrack model introduced by Caroselli [20] is a widely accepted model for emulating medium noise in magnetic recording systems. The term "medium noise" refers to the combination of all disturbances that are caused by the magnetic medium during the non-ideal write/read process. We refer to *medium noise* as the combination of *transition noise* (consisting of position jitter and read-back pulse widening), partial signal erasure (PSE), and non-linear transitions shifts (NLTS). These disturbances (and others as well) can be modeled by the microtrack model.

Transition Noise

Now, we assume that the transition-width parameter a is greater than zero (see also Section 2.2) and that the average magnetization profile $m_0(\zeta)$ exhibits a *tanh*-like shape. The actual magnetization profile will vary from transition to transition such that each read-back pulse will differ from the average read-back pulse. The difference is referred to as *transition noise*. It consists of position jitter and pulse widening. Position jitter refers to a shift in the position of the read-back pulse. Pulse widening refers to redistribution of the energy of the pulse towards the sides and away from the center [20].

The cumulative probability distribution for a change in the magnetization along a track is obtained by scaling and shifting the average magnetization profile (2.52), i.e.

$$Pr(\zeta' \leq \zeta) = \frac{1}{2}\left(1 + \tanh\frac{2\zeta}{a\pi}\right). \tag{4.16}$$

Through differentiation, we obtain the probability density of a change in magnetization as

$$p_{\mathrm{J}}(\zeta) = \frac{M_{\mathrm{r}}}{\pi a}\mathrm{sech}^2\left(\frac{2\zeta}{\pi a}\right). \tag{4.17}$$

We call $p_{\mathrm{J}}(\cdot)$ the *jitter distribution*. For sake of simplicity, we approximate[1] the squared *sech*-pdf of the jitter by a Gaussian pdf as in [20] with variance

$$\sigma_{\mathrm{J}}^2 = \frac{\pi}{2}\cdot a^2. \tag{4.18}$$

[1]This approximation is widely accepted [20].

A Gaussian jitter pdf corresponds to an average magnetization profile with a shape like an error-function.

Channel Model

Medium noise is data-dependent and results from the random microstructure of the grains in thin-film recording media. The microtrack model imitates the random zig-zag transition effects. It is specified by the number of microtracks N, the transition-width parameter a, and the threshold L_{th}, below which two transitions erase each other[2]. The random zig-zag form of a transition is captured by dividing the recording track into N equally-sized microtracks. In Fig. 4.23 the microtrack model is shown as described in [36] with four microtracks. An ideal transition exhibits an average magnetization profile with a shape of a step function. Such ideal transitions are written on each microtrack at a position shifted randomly from the ideal location of the overall transition. The noiseless output of the magnetic recording channel to a

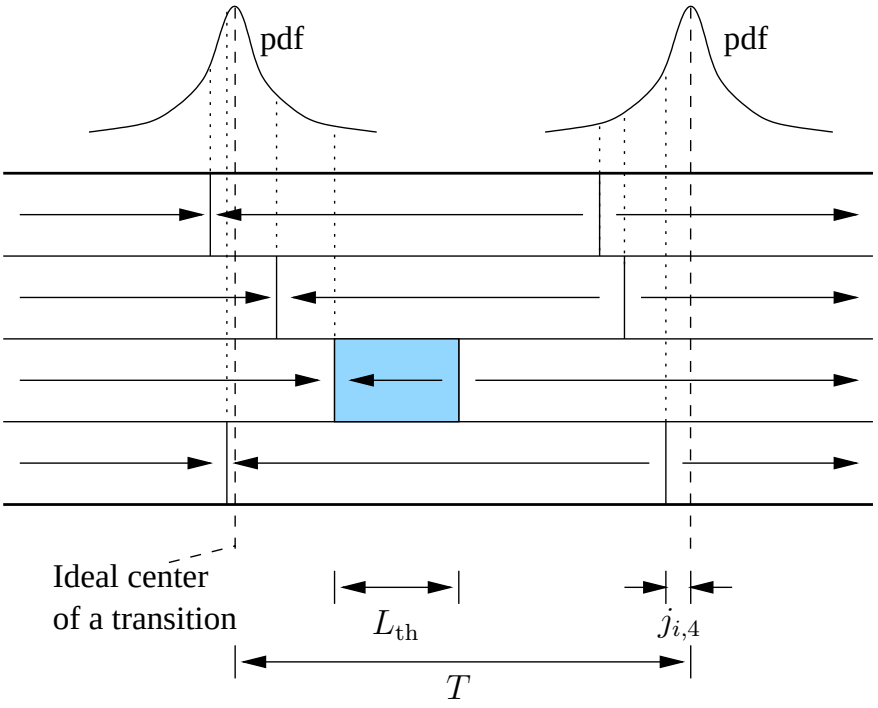

Figure 4.23: The microtrack model.

single positive transition is then given by

$$v(t) = \frac{1}{N} \sum_{i=1}^{N} g(t - J_i), \tag{4.19}$$

[2]If desired, the microtrack model can be refined to incorporate overwrite effects, MR head read-back nonlinearities, write precompensation, etc. [20].

where $g(\cdot)$ is the Lorentzian pulse. Note that $v(t)$ is a random variable. The jitter process $\{j_i\}$ is modeled as i.i.d. process distributed according to $p_{\mathrm{J}}(\cdot)$.

The noiseless output $v(t)$ is therefore given by

$$v(t) = \frac{1}{N} \sum_{k=-\infty}^{+\infty} v_k \sum_{i=1}^{N} g(t - kT - J_{i,k}), \tag{4.20}$$

where $v_k = (x_k - x_{k-1})/2$, with initial condition $x_{-1} = -1$. The x_k's are bipolar and i.u.d.. Hence, $v_k \in \{-1, 0, +1\}$, and the transition process $\{v_k\}$ is a correlated process. Moreover, $J_{i,k}$ is the jitter random variable of the i-th microtrack at the k-th time step. Note that (4.20) is a time-varying convolution. The noisy output $y(t)$ is given by the noiseless output $v(t)$ that is corrupted by AWGN, $n(t)$. The AWGN is determined by its one-sided power spectral density N_0 and represents electronics noise.

In summary, the behavior of our model for the magnetic recording channel is specified by the five parameters PW50$/T, a, N, L_{\mathrm{th}}$, and $N_0/2$ (see Fig. 4.24). The low-pass filter models the finite bandwidth in the receiver.

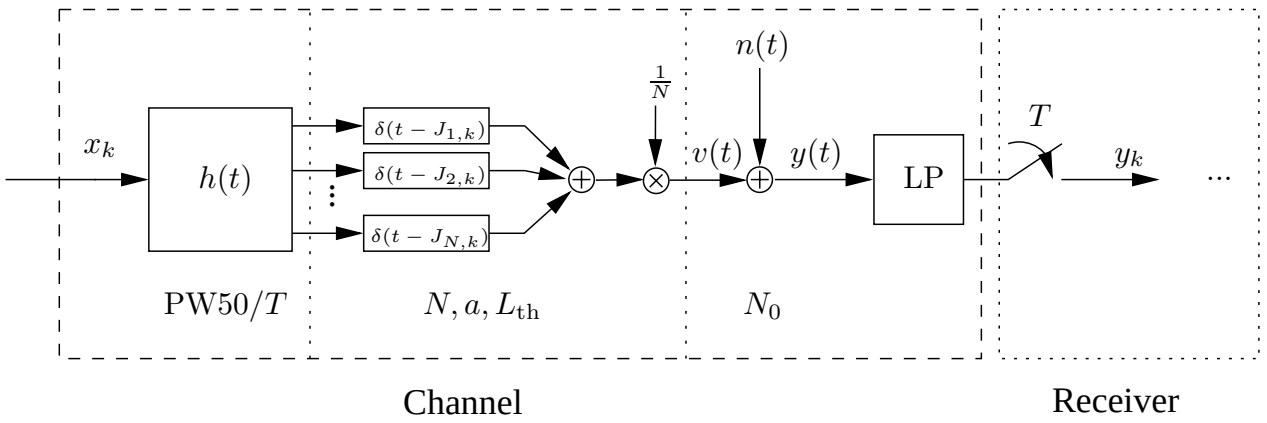

Figure 4.24: A model with five parameters for the magnetic recording channel. Note: $h(t) = \frac{1}{2} \cdot \big(g(t) - g(t - T)\big)$.

Medium Noise: Ideal, Smooth, and Real Transitions

The microtrack model promises to analyze write-head and transition noise separately [20]. This separation is a consequence of the definition of *transition* noise. We will adopt the definition of transition noise given by Caroselli in his thesis [20]: For a given jitter distribution, transition noise is the difference between a finite and an infinite number of microtracks.

For infinitely many microtracks, the noiseless output to a single transition at the input becomes

$$v|_{N=\infty}(t) = (p_{\mathrm{J}} * g)(t), \tag{4.21}$$

where $*$ denotes the convolution operator. Using Taylor expansion, we obtain

$$v|_{N=\infty}(t) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \mathrm{E}_{p_{\mathrm{J}}}[J^k] \frac{\partial^k}{\partial t^k} g(t). \tag{4.22}$$

On the other hand, applying Taylor expansion to (4.19), we obtain, for a finite number of microtracks,

$$v(t) = g(t) - \sum_{i=1}^{N} \frac{J_i}{N} g'(t) + \sum_{i=1}^{N} \frac{J_i^2}{2N} g''(t) \pm \ldots. \tag{4.23}$$

The difference between a finite and infinite number of microtracks therefore equals a weighted sum of the derivatives of the Lorentzian. These weights are the differences between the ensemble moments and the estimated moments (by N samples) of the jitter pdf. For a single transition, this amounts to

$$n_{\mathrm{t}}(t) = v(t) - v|_{N=\infty}(t) \tag{4.24}$$

$$= -\left[\sum_{i=1}^{N} \frac{j_i}{N} - \mathrm{E}[j]\right] g'(t) + \frac{1}{2}\left[\sum_{i=1}^{N} \frac{j_i^2}{N} - \mathrm{E}[j^2]\right] g''(t) \pm \ldots \tag{4.25}$$

where $n_{\mathrm{t}}(t)$ stands for transition noise. Medium noise is obtained if there are several transitions interacting with each other. Through these interactions, the realizations j_i are further disturbed by PSE and NLTS.
With increasing N, the estimated moments in (4.25) become more accurate, and the differences start vanishing. For today's products, realistic numbers for N range from 10 to 50.

With this notion of transition noise in mind, we can distinguish three different medium magnetization profiles (2.52) and consequently three different *transitions* (see Fig. 4.25):

1. Ideal transition: An ideal transition has width zero, i.e. $a = 0$. The jitter variance (4.18) is zero. There is no jitter, and all magnetic particles change their polarization at the same location. There is no transition noise and no medium noise provided that $L_{\mathrm{th}} < T$ and there are no NLTSs.

2. Smooth transition: A smooth transition has an infinite number of microtracks, $N = \infty$, but a finite transition-width parameter a. It exhibits a *smooth* average magnetization profile that in our case is shaped like an error-function. The slope at the origin is determined by the inverse of the a-parameter. The steepness of the slope is a measure of the quality of the write head. A large a signifies a wide transition.

3. Real transition: A real transition has a *finite* number of microtracks N and a finite transition-width parameter a. Owing to the granularity of the medium, it exhibits a *bumpy*, error-function-like magnetization profile. The thinner the granularity of the medium, the smoother the transition and the more microtracks are needed to model this transition.

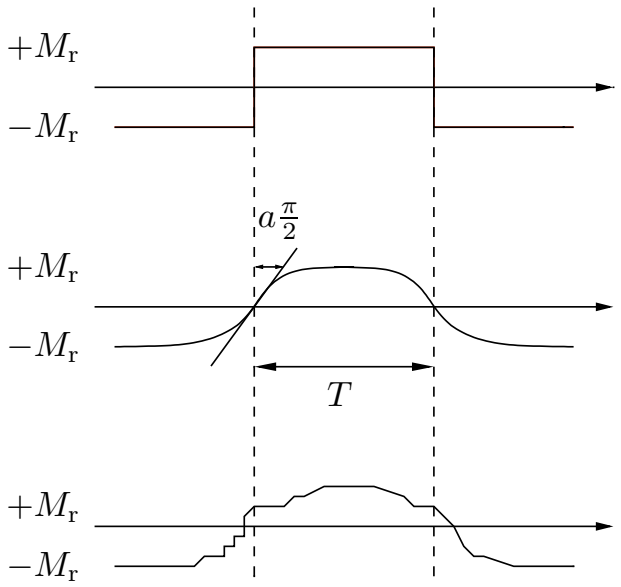

Figure 4.25: Average magnetization profiles: ideal transition (top), smooth transition (middle), and real transition (bottom). The positive and negative remanent state of the medium magnetization are indicated by $+M_\mathrm{r}$ and $-M_\mathrm{r}$ respectively.

4.2.2 Information Rates from FSMs

Implementing the Microtrack Model

The Gaussian jitter distribution was truncated at the adjacent transition locations; i.e. $|j_{i,k}| \leq T$. This truncation allows to check for partial signal erasure with the help of a four-state trellis similarly to the trellis of the binary jitter channel (see next subsection).

An FSM with memory ten was chosen. From the previous subsection, we know that this is sufficient to accurately bound the linear Lorentizan channel (see also Fig. 4.16). The parameters of the microtrack model are listed in Table 4.2 together with the PSE and NLTS. The percentage of PSE is in good agreement with the theoretical results for an isolated dibit [20]. This is due to truncating the Gaussian distribution to the adjacent transition locations, such that most partial signal erasures originate from isolated dibits. NLTSs were implemented as

described by Caroselli [20] with transition interaction length of three. The percentage of NLTS was restricted to be not more than ten percent of the bit spacing T parameter as NLTS can in practice be mitigated by pre-write compensation. Note that these percentages depend neither on PW50/T nor on N, but on a and L_{th}.

Parameter	Numerical value
PW50/T	$2.0, 3.2$
a	$0.3\ T$
L_{th}	a
N	$1, 2, 4, 8, 16, 32$
Measured quantities	Numerical value
Percentage of PSE	9.51%
Percentage of NLTS	1.93%

Table 4.2: Parameters of the microtrack model.

Definitions of Medium Noise Power, Medium Noise Factor, and SNR

We would like to compare different noise blends, while keeping the total noise power constant. This approach is a bit questionable as medium noise is not Gaussian distributed. Thus, the second moment may not be a good measure for the medium noise distribution. On the other hand, in the today's receiver only the second moment is used for detection.

In order to compare different noise blends, we need to quantify the power of the noise caused by the medium. The medium noise power (MNP) is computed from two independent channel measurements. First, the channel is simulated with the Lorentzian waveform, a fixed a, one microtrack, and without AWGN. The total power of the received signal comprises therefore the signal and the medium noise originating from a single microtrack. This power is denoted by TRP(1). Secondly, the same procedure is executed with an infinite number of microtracks. For practical reasons, this is accomplished using a waveform that was obtained by convoluting the Lorentzian waveform with the jitter distribution. The total power of the received signal (without the AWGN) is denoted by TRP(∞). Note that for both measurements the source and thus the number of transitions is the same.

The *medium noise power* is then the difference between these two received powers, i.e.

$$\text{MNP} \triangleq 2.0 \cdot \big(\text{TRP}(1) - \text{TRP}(\infty)\big) \tag{4.26}$$

where the factor 2.0 comes from the fact that our source is by assumption an i.u.d. source and, thus, only half of the bits cause a transition (experience medium noise). As the amount

of medium noise depends on the number of transitions, the MNP is always a function of the source.
Next, we define the *medium noise factor* (MF) as

$$\mathrm{MF} \triangleq \frac{\mathrm{MNP}/N}{\mathrm{MNP}/N + N_0/2}, \tag{4.27}$$

and note that it depends on the number of microtracks N. MF = 0.1 means 10% medium noise and 90% AWGN for a given number of microtracks.

In practice, we are given a certain operating point, i.e. a particular PW50/T, N, a, L_{th}, and N_0. For our computation, we can artificially generate operating points in the following way. For a given a and a given PW50/T, we first compute the MNP as described above. After having fixed the number of microtracks, N, and the medium noise factor, MF, we solve (4.27) for $N_0/2 = \sigma^2$. Different SNR points are then obtained by varying N. In a computer-simulation environment, N must be a positive integer number.

The channel is normalized to "1" under the assumption that transition noise is absent. Thus, we obtain the following SNR definition

$$\mathrm{SNR} \triangleq \frac{\mathrm{TRP}(\infty)}{N_0/2 + 0.5 \cdot \mathrm{MNP}/N} \tag{4.28}$$

with the constraints that (4.27) is fulfilled and that N is a positive integer number. The factor 0.5 in the denumerator is needed to compensate for the factor 2.0 in (4.26).

Numerical Results

Fig. 4.26 shows the Lorentzian step response of the trained FSM in the absence of (MF=0.0) and in the presence of medium noise (MF=0.96). For illustrative purposes, the latter two curves are shifted by one tap to the right. In both noise scenarios, we recognise the Lorentzian pulse. It is symmetric in the absence of medium noise and a bit skewed in the presence of medium noise. The impulse response of the FSM, i.e. the response to two consecutive transitions, is shown in Fig. 4.27. We observe that a broader Lorentzian pulse, i.e. PW50/T = 3.2, leads to a smaller but wider impulse response. In the medium noise scenario, the impulse responses become a bit skewed. In Fig. 4.27, they are shifted one tap to the right for illustrative purposes.

Fig. 4.28 shows a histogram of received samples associated with the transition from state $s_{k-1} = (-1,-1,-1,-1,+1,-1,-1,-1,-1,-1)$ at time $k-1$ to the time-k state $s_k = (-1,-1,-1,-1,-1,+1,-1,-1,-1,-1)$ in our FSM. A Lorentzian pulse with PW50/T = 2.0

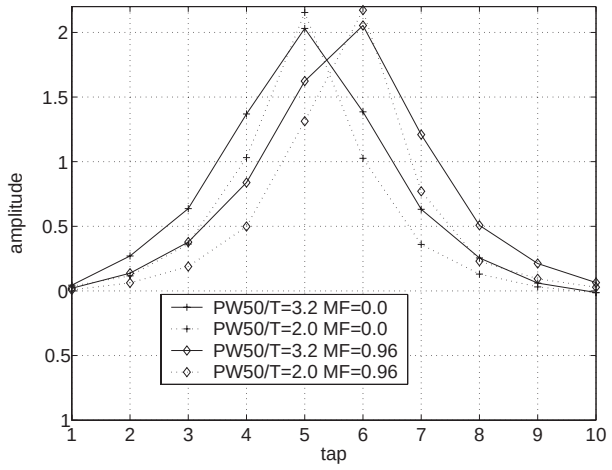

Figure 4.26: Step response of the FSM.

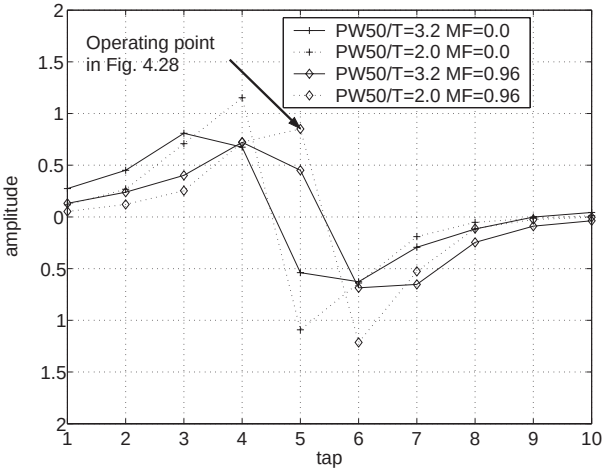

Figure 4.27: Impulse response of the FSM.

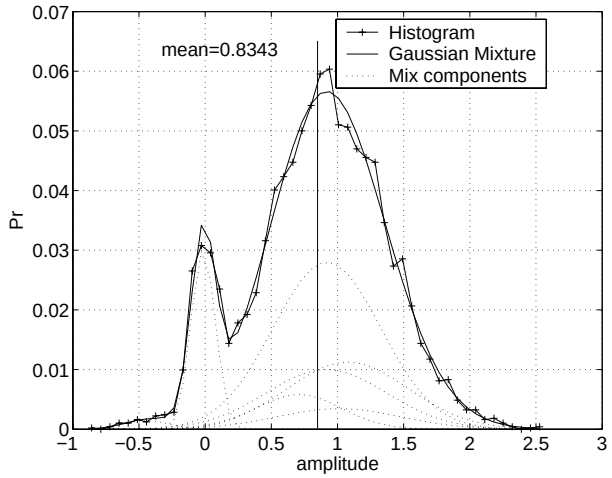

Figure 4.28: Histogram of a transition.

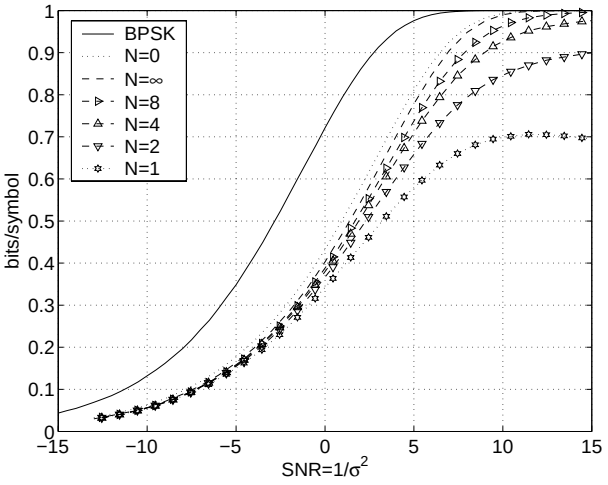

Figure 4.29: Influence of N on $\underline{I}_{\mathrm{a}}(\mathcal{X};\mathcal{O})$.

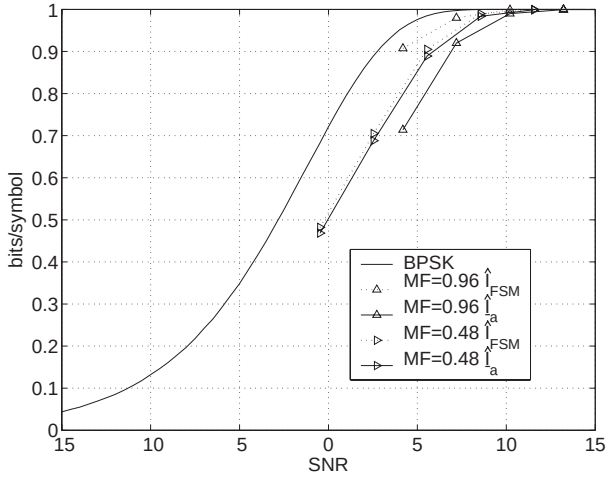

Figure 4.30: Estimated information rates for PW50/$T = 2.0$.

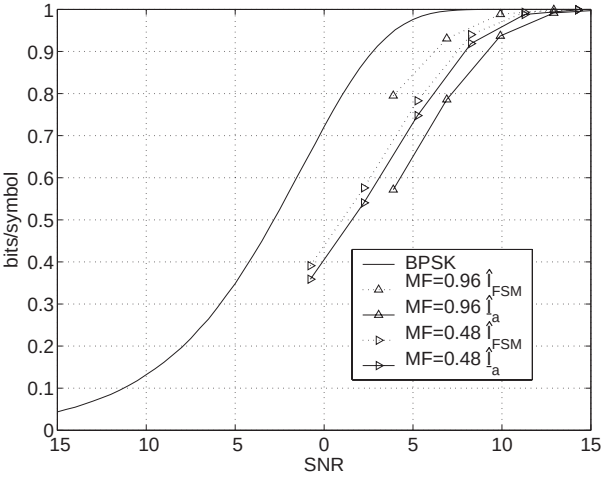

Figure 4.31: Estimated information rates for PW50/$T = 3.2$.

was selected. Differentiating s_{k-1} gives us $(0, 0, 0, 0, +1, -1, 0, 0, 0, 0)$, the ideal read-back waveform, i.e. no medium noise and a Lorentzian pulse with PW50$/T = 0.0$ (=Dirac function) as step response. On the branch connecting state s_{k-1} to state s_k, we can thus collect samples that originate from two overlapping Lorentzian pulses; one centered around tap "5" and one centered around tap "6". This operating point is indicated in Fig. 4.27 with an arrow. The mean of the histogram in Fig. 4.28 corresponds very well to that point.
We clearly see that the histogram does not exhibit the shape of a Gaussian distribution. The effect of PSE is responsible for the elevation around 0, i.e. some transitions are erased. From Fig. 4.28, we infer by inspection that the histogram is well approximated by a Gaussian mixture consisting of $\xi = 8$ single Gaussian distributions. As the FSM consists of $L = 1024$ states, there are in total $2 \cdot 1024 \cdot (8 + 8 + 7)$ parameters to be trained with observations generated by the microtrack model.

In Fig. 4.29, estimates of the lower bound $\underline{I}_{\mathrm{a}}(\mathcal{X}; \mathcal{O})$ are shown vs. SNR $= 1/\sigma^2$, the variance of the AWGN. Here, a Lorentzian pulse with PW50$/T = 3.2$ was chosen. For $N = 1$, the curve saturates around 0.7 bits/symbol. This comes from the fact that by incresaing the SNR, we reduce that amount of AWGN, but the amount of medium noise remains unchanged. By increasing N, the amount of medium noise is decreased and all curves approach the one for $N = \infty$. As outlined above, the information rate for $N = \infty$ was obtained by first convoluting the jitter distribution with the Lorentizan pulse and then computing the information rate for this new waveform. Due to the convolution with the jitter distribution, the resulting waveform is broader and leads to an information rate that lies slightly to the right of the "pure" Lorentzian pulse (indicated by $N = 0$ in Fig. 4.29)

Fig. 4.30 shows two pairs of curves: Estimates of the lower bound $\underline{I}_{\mathrm{a}}(\mathcal{X}; \mathcal{O})$ (solid lines) and estimates of the FSM information rate which is denoted by $I_{\mathrm{FSM}}(\mathcal{X}; \mathcal{Y})$ (dotted lines). The discrete operating points correspond to $N = 1, 2, 4, 8, 16, 32$. To compute $I_{\mathrm{FSM}}(\mathcal{X}; \mathcal{Y})$, we first generate a long output sequence y^n of the trained FSM and then execute the forward-only method on the trellis induced by the FSM (see also Fig. 3.2). Note that $I_{\mathrm{FSM}}(\mathcal{X}; \mathcal{Y})$ is *not* a lower bound on $I(\mathcal{X}; \mathcal{O})$. If the FSM is a perfect replica of the channel law of the original channel, then $I_{\mathrm{FSM}}(\mathcal{X}; \mathcal{Y}) = \underline{I}_{\mathrm{a}}(\mathcal{X}; \mathcal{O}) = I(\mathcal{X}; \mathcal{Y})$. Thus, we may conjecture that $I_{\mathrm{FSM}}(\mathcal{X}; \mathcal{Y})$ is a lower bound on $I(\mathcal{X}; \mathcal{O})$, but we cannot prove it.
Referring to Fig. 4.30, we note that for both pairs of curves different noise blends were used; namely MF= 0.48 and MF= 0.96 with PW50$/T = 2.0$. The upper bound of (3.141) simply neglects the amount of medium noise in the total noise (as we have no knowledge of its distribution). It is thus quite lose and, hence, was not plotted.
We observe that the gap between the FSM information rate $I_{\mathrm{FSM}}(\mathcal{X}; \mathcal{Y})$ and the lower bound $\underline{I}_{\mathrm{a}}(\mathcal{X}; \mathcal{O})$ increases with increasing amount of medium noise. The FSM information rate sug-

gests that medium noise is perferable to AWGN whereas the lower bound seems to favor AWGN. The same can also be seen for PW50/T=3.2 in Fig. 4.31. The difference of these two bounds manifestates the mismatch between the FSM and the channel law of the original channel. Thus, the only (unfortunately) conclusion we can draw is that our FSM is a bad fit to the observed data. A better model is needed to answer the question whether medium noise is preferable to AWGN or not.

4.2.3 The Binary Jitter Channel

The binary jitter channel (BJC) [4] is motivated by the expectation that in future high-density magnetic recording devices the noise of the magnetic medium will dominate other noise sources. This "medium noise" is signal dependent and comes in two quite different forms: first isolated transitions (i.e. changes of magnetic polarization) are affected by *jitter*: the transition is read at a different position than where it was written. Secondly, very short polarization regions tend to be *unstable*: the transitions move towards and cancel each other. Thus, the BJC is a simplistic version of the microtrack model.

The BJC is related to the so-called bit-shift or peak-shift magnetic recording channel because they both model the jitter effect. Upper and lower bounds on the capacity of the bit-shift channel were presented by Shamai et al. in [69] and by Baggen in [7]. These authors use a combinatorial approach. The BJC differs however from the bit-shift channel in that it additionally incorporates the erasure effect. It does therefore affect the number of ones (representing transitions) upon transmission, i.e. the number of ones in the output sequence Y^n may be less than in the input sequence X^n.

Description of the BJC

Let $X_k \in \{0,1\}$ and $Y_k \in \{0,1\}$ be the time-k input and output, respectively, of the BJC, where 1 stands for a transition and 0 stands for no transition. Hence, $X_k = 1$ ($Y_k = 1$) means a transition was written in (read from) the time-k slot. The BJC $X_k \to Y_k$ is decomposed into a memoryless probabilistic channel $X_k \to J_k$ and a deterministic channel $J_k \to Y_k$ with memory. The auxiliary random variable J_k takes values in the set $\{0\} \cup \{D^i : i = -m, -m+1, \ldots, m\}$ for some positive integer m; $J_k = D^i$ means that a transition written into the time-k slot and was moved into slot $k+i$ (in contrast to the jitter variable $J_{i,k}$ in the microtrack model, J_k is a discrete random variable). We consider here the simplest case with $m = 1$ such that the

probabilistic channel without memory $X_k \to J_k$ is given by

$$p_{J|X}(j_k|0) = \begin{cases} 1 & \text{if } j_k = 0 \\ 0 & \text{else} \end{cases} \tag{4.29}$$

and

$$p_{J|X}(j_k|1) = \begin{cases} 0 & \text{if } j_k = 0 \\ 1-2p & \text{if } j_k = 1 \\ p & \text{if } j_k = D \text{ or } j_k = D^{-1} \end{cases} . \tag{4.30}$$

The deterministic channel with memory $J_k \to Y_k$ is responsible for the "sorting". This "sorting" is determined by the underlying physical phenomena and is given for a magnetic recording channel by the trellis structure in Fig. 4.32. The states are labeled with their zero-input response, and the branches are labeled with pairs of an input $D \cdot J_{k+1}$ and an output Y_k. The use of future inputs J_{k+1} makes the trellis causal. An example for the input, state, jitter, and output sequence is given in Tab. 4.3.

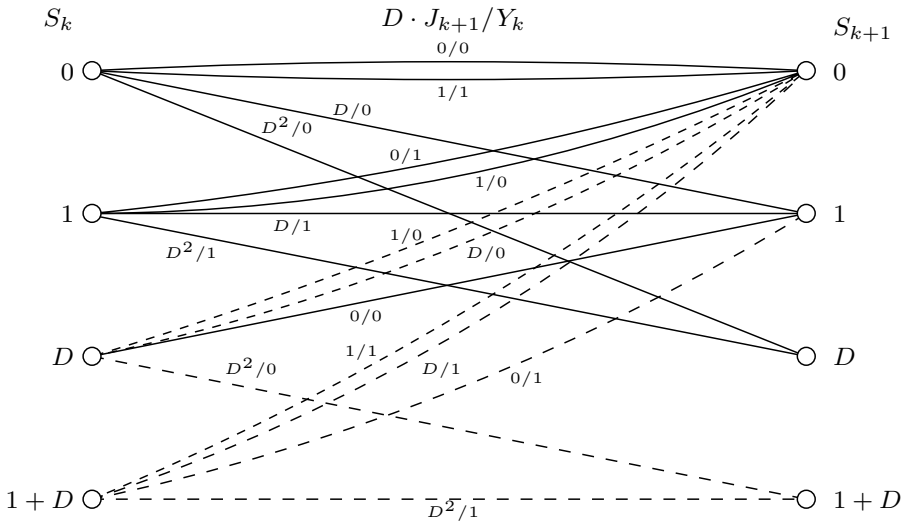

Figure 4.32: Trellis of the BJC. The dashed lines do not exist if a $(1, \infty)$-RLL constraint is imposed on the input.

Information Rates of the BJC

We consider now the case where a $(1, \infty)$-RLL source is connected to a BJC with $m = 1$. A $(1, \infty)$-RLL source requires that each 1 is followed by at least one 0 and can be realized with two (minimal state-space realization) or more states.

Time	-1	0	1	2	3	4	5	6	7	8	9 ...
Input X_k		0	1	1	1	0	1	0	1	0	1 ...
Jitter J_k		0	D	D^2	1	0	D	0	1	0	D^2 ...
State S_k	0	0	1	D	0	0	1	0	0	0	D ...
Output Y_k	0	0	1	0	0	0	1	1	0	0	...

Table 4.3: An example for the BJC with input, state, jitter, and output sequence. An erasure occurred between X_2 and X_3.

An estimate of the output entropy rate $h(\mathcal{Y})$ is computed with the forward sum-product algorithm operating on the joint source/channel trellis, i.e. the $(1,\infty)$-RLL source together with BJC. This joint trellis is shown in Fig. 4.33, where the state S_k is obtained by concatenating a state of the $(1,\infty)$-RLL source, S_k^{RLL}, with a state of the BJC, S_k^{BJC}, in the following way:

$$S_k = (S_k^{\mathrm{RLL}}, S_k^{\mathrm{BJC}}), \tag{4.31}$$

with $S_k^{\mathrm{RLL}} \in \{1,2\}$ and $S_k^{\mathrm{BJC}} \in \{0,1,D,1+D\}$. In Fig. 4.33, the states associated with state

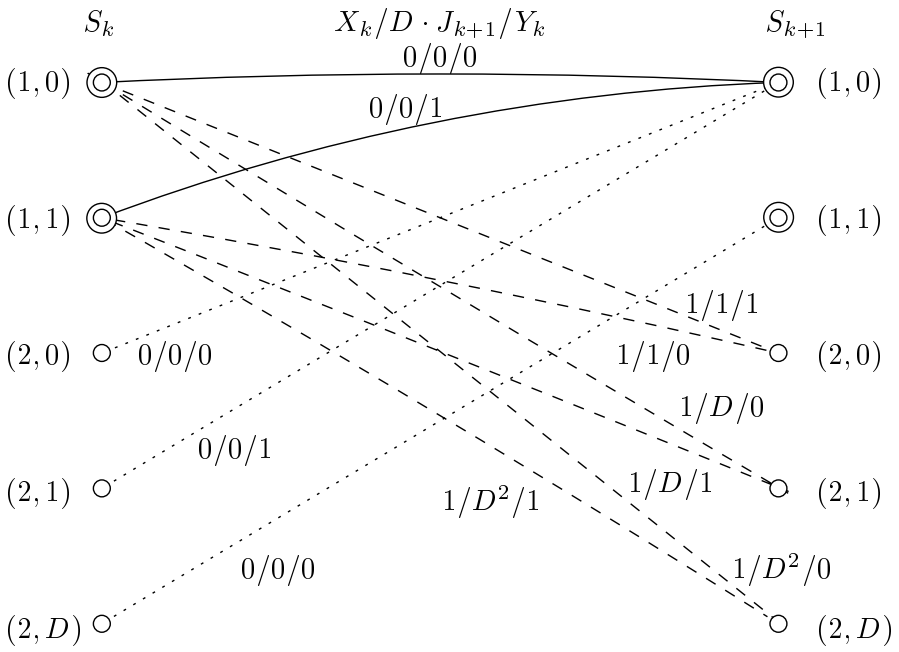

Figure 4.33: Joint trellis of the $(1,\infty)$-RLL source together with the BJC.

"1" of the $(1,\infty)$-RLL source are depicted with double circles; the other states belong to state "2"of the $(1,\infty)$-RLL source.

An estimate of $h(\mathcal{Y}|\mathcal{X})$ is computed on a reduced trellis induced by the input sequence x^n as described in Chapter 3.

Lower Bounds on Capacity

The optimization of the state-transition probabilities is performed only on the STPs of the $(1,\infty)$-RLL source, as the state-transition probabilities of the BJC are given by the jitter probability and cannot be changed. This means that the noisy adjacency matrix, as defined in (3.70) is a two-by-two matrix.

Referring to Fig. 4.33, the state-transition probabilities leaving state $S_k = (1,0)$ are given by

$$\mathbf{Q}\big((1,0),(1,0)\big) = \mathbf{Q}^{\mathrm{RLL}}(1,1) \tag{4.32}$$
$$\mathbf{Q}\big((1,0),(2,0)\big) = \mathbf{Q}^{\mathrm{RLL}}(1,2)\cdot p_{\mathrm{J}} \tag{4.33}$$
$$\mathbf{Q}\big((1,0),(2,1)\big) = \mathbf{Q}^{\mathrm{RLL}}(1,2)\cdot (1-2p_{\mathrm{J}}) \tag{4.34}$$
$$\mathbf{Q}\big((1,0),(2,D)\big) = \mathbf{Q}^{\mathrm{RLL}}(1,2)\cdot p_{\mathrm{J}} \tag{4.35}$$

and the ones leaving state $S_k = (1,1)$ by

$$\mathbf{Q}\big((1,1),(1,0)\big) = \mathbf{Q}^{\mathrm{RLL}}(1,1) \tag{4.36}$$
$$\mathbf{Q}\big((1,1),(2,0)\big) = \mathbf{Q}^{\mathrm{RLL}}(1,2)\cdot p_{\mathrm{J}} \tag{4.37}$$
$$\mathbf{Q}\big((1,1),(2,1)\big) = \mathbf{Q}^{\mathrm{RLL}}(1,2)\cdot (1-2p_{\mathrm{J}}) \tag{4.38}$$
$$\mathbf{Q}\big((1,1),(2,D)\big) = \mathbf{Q}^{\mathrm{RLL}}(1,2)\cdot p_{\mathrm{J}}. \tag{4.39}$$

The state-transition probabilities of the other three states equal 1, or to be precise

$$\mathbf{Q}\big((2,0),(1,0)\big) = \mathbf{Q}^{\mathrm{RLL}}(2,1) \tag{4.40}$$
$$\mathbf{Q}\big((2,1),(1,0)\big) = \mathbf{Q}^{\mathrm{RLL}}(2,1) \tag{4.41}$$
$$\mathbf{Q}\big((2,D),(1,1)\big) = \mathbf{Q}^{\mathrm{RLL}}(2,1). \tag{4.42}$$

Similarly as in Chapter 3, we introduce the following shorthand notations

$$v_k^{\mathrm{RLL}}(i,j|Y^n) \triangleq Pr(S_{k-1}^{\mathrm{RLL}} = i, S_k^{\mathrm{RLL}} = j|Y^n) \tag{4.43}$$
$$v_{k-1}^{\mathrm{RLL}}(i|Y^n) \triangleq Pr(S_{k-1}^{\mathrm{RLL}} = i|Y^n) \tag{4.44}$$
$$v_k^{\mathrm{RLL}}(i,\ell,j|Y^n) \triangleq Pr(S_{k-1}^{\mathrm{RLL}} = i, B_k = \ell, S_k^{\mathrm{RLL}} = j|Y^n) \tag{4.45}$$
$$v_{k-1}^{\mathrm{RLL}}(i,\ell|Y^n) \triangleq Pr(S_{k-1}^{\mathrm{RLL}} = i, B_k = \ell|Y^n) \tag{4.46}$$

with

$$|\mathsf{B}_k| = \begin{cases} 2 & \text{if } \big(S_{k-1}^{\mathrm{RLL}} = 1\big) \cap \big(S_k^{\mathrm{RLL}} = 1\big) \\ 6 & \text{if } \big(S_{k-1}^{\mathrm{RLL}} = 1\big) \cap \big(S_k^{\mathrm{RLL}} = 2\big) \\ 3 & \text{if } \big(S_{k-1}^{\mathrm{RLL}} = 2\big) \cap \big(S_k^{\mathrm{RLL}} = 1\big) \end{cases}. \tag{4.47}$$

This grouping of the branches is also visualized in Fig. 4.33: the solid, the dashed, and the dotted lines are assigned to the $(1,1)$, $(1,2)$, and $(2,1)$ transition in the $(1,\infty)$-RLL source.

The a-posteriori state-transition weights of the $(1,\infty)$-RLL source are now expressed as follows

$$\begin{aligned}\mathbf{T}^{\mathrm{RLL}}(i,j) &= \lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\mathrm{E}_{p_{Y^n|S_k^{\mathrm{RLL}}=j,S_{k-1}^{\mathrm{RLL}}=i}}\left[\log_2\frac{v_k^{\mathrm{RLL}}(i,j|Y^n)}{v_{k-1}^{\mathrm{RLL}}(i|Y^n)}\right] \qquad (4.48)\\ &= \lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\mathrm{E}_{\sum_\ell p_{Y^n,B_k=\ell|S_k^{\mathrm{RLL}}=j,S_{k-1}^{\mathrm{RLL}}=i}}\left[\log_2\frac{v_k^{\mathrm{RLL}}(i,j|Y^n)}{v_{k-1}^{\mathrm{RLL}}(i|Y^n)}\right]\\ &= \lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\left[\sum_\ell Pr(B_k=\ell|S_k^{\mathrm{RLL}}=j,S_{k-1}^{\mathrm{RLL}}=i)\right.\\ &\qquad \left.\mathrm{E}_{p_{Y^n|S_k^{\mathrm{RLL}}=j,S_{k-1}^{\mathrm{RLL}}=i,B_k=\ell}}\left[\log_2\frac{v_k^{\mathrm{RLL}}(i,\ell,j|Y^n)}{v_{k-1}^{\mathrm{RLL}}(i,\ell|Y^n)}\right]\right] \qquad (4.49)\end{aligned}$$

as the last line follows from conditioning the expectation on a particular state-transition. Thus the expectation term in (4.49) equals the a-posteriori state-transition weight of the joint source/channel trellis, i.e.

$$\mathbf{T}(i,\ell,j) = \lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\mathrm{E}_{p_{Y^n|S_k^{\mathrm{RLL}}=j,S_{k-1}^{\mathrm{RLL}}=i,B_k=\ell}}\left[\log_2\frac{v_k^{\mathrm{RLL}}(i,\ell,j|Y^n)}{v_{k-1}^{\mathrm{RLL}}(i,\ell|Y^n)}\right]. \qquad (4.50)$$

This allows us to obtain the a-posteriori state-transition weight of the RLL-source in the following form

$$\mathbf{T}^{\mathrm{RLL}}(i,j) = \sum_\ell Pr(B_1=\ell|S_1^{\mathrm{RLL}}=j,S_0^{\mathrm{RLL}}=i)\mathbf{T}(i,\ell,j) \qquad (4.51)$$

where the probability term is independent of time.

For finite n, we obtain the empirical counterpart of the a-posteriori state-transition weights of (4.50) as follows

$$\hat{\mathbf{T}}(i,\ell,j) = \frac{1}{n(i,\ell,j)}\sum_{\substack{k=1\\(S_{k-1}=i,B_k=\ell,S_k=j)\in x^n}}^{n}\log_2\frac{v_k^{\mathrm{RLL}}(i,\ell,j|Y^n)}{v_{k-1}^{\mathrm{RLL}}(i,\ell|Y^n)} \qquad (4.52)$$

where $n(i,\ell,j)$ is the number of state-transitions from state i to state j over branch ℓ. The conditioned version of the a-posteriori state-transition weights is advantageous here, as the probability of observing a channel output is either one or zero given a received symbol. Thus the empirical counterpart to (4.51) becomes to

$$\hat{\mathbf{T}}^{\mathrm{RLL}}(i,j) = \frac{1}{\sum_{\ell'} n(i,\ell',j)}\sum_\ell n(i,\ell,j)\hat{\mathbf{T}}(i,\ell,j). \qquad (4.53)$$

The optimized state-transition probabilities $\mathbf{Q}^{\mathrm{RLL}}(i,j)$ of the $(1,\infty)$-RLL trellis are obtained now by the update formula in Theorem 3.5.

Numerical Results

In Fig. 4.34, estimates of the information rates of the BJC driven by a $(1,\infty)$-RLL source (two state realization) are plotted against a varying jitter probability $p_{\mathrm{J}}(\cdot)$ and varying $\mathbf{Q}^{\mathrm{RLL}}(1,1)$, the STP of the $(1,\infty)$-RLL source. Estimates of $H(\mathcal{X}|\mathcal{Y})$ are shown in Fig. 4.35. We see that for each jitter probability, both the information rate as well $H(\mathcal{X}|\mathcal{Y})$ are concave functions of the STP. Thus, the iterative information rate maximization method from Chapter 3 will find the globally optimal $\mathbf{Q}^*(1,1)$. This can be verified in Fig. 4.36 where $\mathbf{Q}^*(1,1)$ was computed by brute-force computation and by the iterative information rate maximization method.

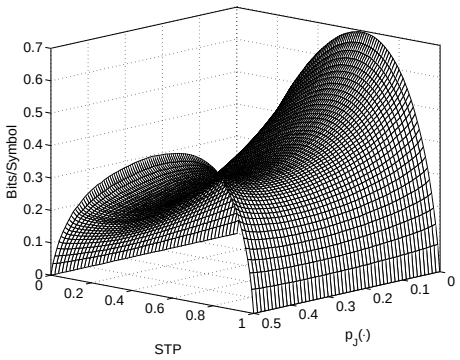

Figure 4.34: Estimated information rates.

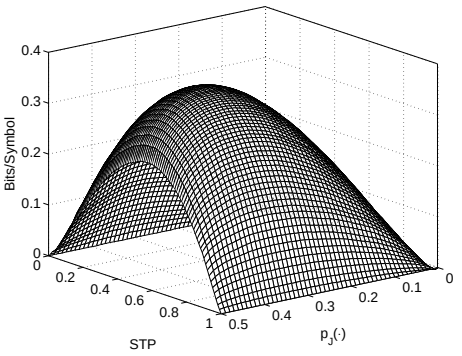

Figure 4.35: Estimated $H(\mathcal{X}|\mathcal{Y})$.

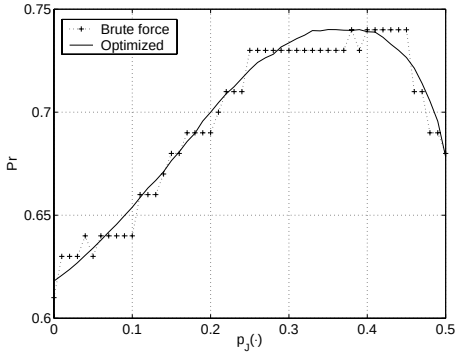

Figure 4.36: Optimal state-transition probabilities $\mathbf{Q}(1,1)$.

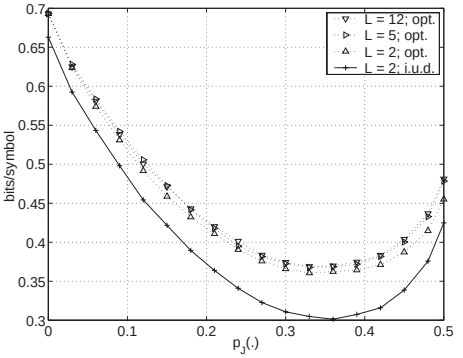

Figure 4.37: Estimated information rates for various input processes.

In Fig. 4.37, estimated information rates are shown for i.u.d. input as well as optimized Markov inputs fulfilling the $(1,\infty)$-RLL constraint. The number $L = 2$ means that a two state realization was used for the $(1,\infty)$-RLL trellis (minimal state-space realization). Non-minimal state-space realization of the $(1,\infty)$-RLL constraint were obtained by considering realizations with $L = 5$ states and $L = 12$ states. In absence of noise, the capacity of $(1,\infty)$-RLL constrained sequences is achieved by the minimal state-space realization. Enlarged state-space realizations provide more state-transition probabilities to be optimized. In the presence

of noise, this leads to higher information rates.

Chapter 5

Summary and Concluding Remarks

The results obtained in this dissertation can be summarized as follows.
Chapter 2:

- The capacity of an irreducible and aperiodic FSM was formulated. We noted that it is simpler to compute the capacity of an irreducible and aperiodic FSM than the capacity of an indecomposable FSC, as the number of constraints imposed on the input process is finite. The capacity of the FSM is a lower bound on the capacity of the corresponding FSC.

Chapter 3:

- We presented a practical sampling-based computation method for computing estimates of entropy rates and consequently information rates of FSMs. Reduced-state versions were also provided. The pivotal observation behind this method is that estimates of the entropy rate of hidden Markov processes can exactly be computed by standard forward sum-product trellis processing of simulated or (in principle) measured channel output data. The method can be described as a stochastic version of the power-method that is routinely used to compute estimates of the largest eigenvalue of non-negative primitive matrices.

- A gradient-based method and Kavčić's method have been examined for maximizing the information rate for a given Markov source model. We showed that by progressively extending the Markov input process over its minimal state-space realization a set of tight lower bounds on capacity is obtained.

- An upper bound on the capacity of FSMs was proposed using the dual expression of the channel capacity. It was shown that this upper bound resides on hidden Markov

entropy rates as well. Thus, the sampling-based method for computing lower bounds on capacity can also be used to compute an upper bound. Moreover, it was noticed that if the lower bound is tight, the upper bound is tight as well.

- Provable upper and lower bounds on the information rate of the discrete memoryless channel were extended to very general ergodic (not necessary finite-state) channels. Estimates of these bounds were computed in Chapter 4 for various channel models using a trained FSM as a computation vehicle. The tightness of the bounds depends on how close the channel law of the auxiliary FSM is to the channel law of the original channel.

Chapter 4:

- For a given trellis length, we computed estimates of information rates for selected channel models assuming bipolar i.u.d. input signals. We observed that the information rate decreases with increasing memory of the channel model; especially at high SNR, where a low noise floor reveals the differences of the channel models.

- Tight lower bounds on capacity of various FSMs were computed. It was found empirically that the iterative information maximization method proposed by Kavčić increases the information rate leading to tight lower bounds. A proof is missing as it could not be shown that the FSM information rate is a concave function over the convex set of state-transition probabilities.
  Owing to the normalization of the channel transfer function, information rates were obtained by this method that exceed in the low SNR regime even the capacity of the memoryless Gaussian channel.

- The effect of medium noise on the information rate was investigated. To this end, FSMs were trained by synthetically generated waveforms (microtrack model). For a fixed noise power, the information rate of the trained FSM was considerably higher for an i.u.d. input process in the medium noise dominated noise scenario than in the AWGN dominated noise scenario. However, these information rates are not provable lower bounds on the information rate of the original microtrack channel. Quite on the contrary, the provable lower bounds suggest that medium noise is detrimental. The discrepancy between these two bounds suggests that a parameterized Gaussian model is a bad model for medium noise (partial erasure).

- A new model for medium noise was developed: the binary jitter channel. It is a nontrivial channel model as it is a nonlinear FSM and the discrete medium noise process contains memory. It models transition shifts and partial signal erasure. Numerical results of tight lower bounds on the capacity are presented for various input processes.

We regret if — after having worked for quite a while on this topic — we have to leave the reader with more questions remaining than there were originally. The good thing though is that the following future research projects seem promising:

- Investigate the iterative information rate maximization method of Kavčić and its realization to the Arimoto-Blahut algorithm in more detail with the objective to proof that it finds a local maximum.
- Devise tight upper bounds in the presence of medium noise. This involves a better understanding of the noise process and a more accurate noise model.
- Find an efficient and practical reduced-state lower bound similar to the reduced-state upper bound.
- Apply the sampling-based computation method to other areas such as multi-input-multi-output channels in wireless communications, Markov random fields and 2-dimensional constraint systems. The information content in expert systems and data-mining algorithms may also be bounded by this method.

Appendix A

A Numerical Example: $(0, 2)$-RLL Source

There are various definitions of run-length-limited (RLL) sequences. We follow the definition of Proakis [61] and define an RLL source in the following way.

Definition A.1 ((d, k)-RLL Source [61]) *A (d, k)-RLL source emits a sequence of 1s and 0s. The parameters d and k are non-negative integers with k always larger than d. The parameter d indicates the minimum number of 0s between two 1s and the parameter k indicates the maximum number of 0s between two 1s.*

In [45], Immink defines RLL sequence via binary (d, k)-sequences. These binary (d, k)-sequences are actually the RLL-sequences of Proakis. For Immink, an RLL-sequence is a bipolar sequence that is obtained after *precoding* a binary (d, k)-sequence. The precoding step[1] maps logical ones (representing the positions of the transitions) of the (d, k)-sequence to transitions, i.e. $+1 \rightarrow -1$ or $-1 \rightarrow +1$. The sequence after the precoding operation is termed RLL-sequence in [45]. It can be verified that such an RLL sequence has the virtue that at least $d + 1$ or at most $k + 1$ consecutive *like* symbols occur. In [46], Immink, Siegel, and Wolf follow Proakis' definition of RLL sequences such that we feel confident with our definition of a (d, k)-RLL source.

A $(d = 0, k = 2)$-RLL source premits therefore at most two consecutive 0s. The state-transition diagram and the corresponding trellis section of a minimal state-space realization are shown in Fig. A.1.

[1] The term "precoding" is confusing, as it is in fact a *postcoding* process (see [45]).

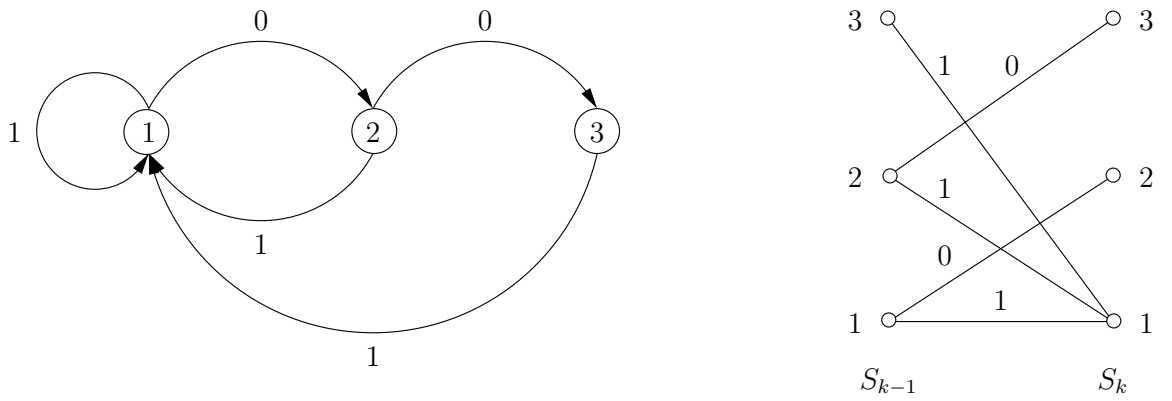

Figure A.1: State-transition diagram and (forward) trellis section of a minimal state-space realization of the $(0, 2)$-RLL source.

A.1 Forward Trellis

The forward adjacency matrix of the minimal state-space realization shown in Fig. A.1 (right part) is

$$\mathbf{A}_\mathrm{f} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} . \tag{A.1}$$

The largest (in terms of magnitude) eigenvalue of $\mathbf{A}$ is $\rho(\mathbf{A}_\mathrm{f}) = 1.8393$. The capacity is $C = \log_2 1.8393 = 0.8791$ bits/symbol. The right and left eigenvectors corresponding to the largest eigenvalue are

$$\mathbf{r}_\mathrm{f}^T = \begin{pmatrix} 0.4196 & 0.3522 & 0.2282 \end{pmatrix}^T \tag{A.2}$$

and

$$\boldsymbol{l}_\mathrm{f} = \begin{pmatrix} 0.5437 & 0.2956 & 0.1607 \end{pmatrix} \tag{A.3}$$

where we normalized $\boldsymbol{r}^T$ and $\boldsymbol{l}$ such that $\sum_i \boldsymbol{r}^T(i) = 1$ and $\sum_i \boldsymbol{l}(i) = 1$ respectively. The entries of the state-distribution vector are given by $\boldsymbol{\pi}_\mathrm{f}(i) = c \cdot \mathbf{r}_\mathrm{f}(i)\boldsymbol{l}_\mathrm{f}(i)$ where c is a normalization constant such that $\sum_i \boldsymbol{\pi}_\mathrm{f}(i) = 1$. The state-distribution vector becomes then

$$\boldsymbol{\pi}_\mathrm{f} = \begin{pmatrix} 0.6184 & 0.2822 & 0.0994 \end{pmatrix} . \tag{A.4}$$

According to Shannon [70], the optimal forward state-transition probabilities are given as follows

$$\mathbf{Q}_\mathrm{f}^*(i,j) = \begin{cases} \frac{r_\mathrm{f}(j)}{r_\mathrm{f}(i)} \frac{1}{\rho(\mathbf{A}_\mathrm{f})} & \forall (i,j) : \mathbf{A}_\mathrm{f}(i,j) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{A.5}$$

and are collected in the forward state-transition probability matrix $\mathbf{Q}^*$, i.e.

$$\mathbf{Q}_{\mathrm{f}}^* = \begin{pmatrix} 0.5437 & 0.4563 & 0 \\ 0.6478 & 0 & 0.3522 \\ 1 & 0 & 0 \end{pmatrix} . \tag{A.6}$$

As $\mathbf{Q}_{\mathrm{f}}$ is a stochastic matrix (the row entries sum up to 1), the largest eigenvalue is one and the corresponding right eigenvector, $\check{\boldsymbol{r}}_{\mathrm{f}}^T$, uniform. The corresponding left eigenvector, $\check{\boldsymbol{l}}_{\mathrm{f}}$, is not uniform but equals the state-distribution vector $\boldsymbol{\pi}_{\mathrm{f}}$. Thus,

$$\check{\boldsymbol{r}}_{\mathrm{f}} = \frac{1}{3} \cdot \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \tag{A.7}$$

and

$$\check{\boldsymbol{l}}_{\mathrm{f}} = \begin{pmatrix} 0.6184 & 0.2822 & 0.0994 \end{pmatrix} = \boldsymbol{\pi}_{\mathrm{f}} . \tag{A.8}$$

A.2 Backward Trellis

The backward adjacency matrix is obtained by transposing the forward adjacency matrix, i.e.

$$\mathbf{A}_{\mathrm{b}} = \mathbf{A}_{\mathrm{f}}^T = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \tag{A.9}$$

and thus $\rho(\mathbf{A}_{\mathrm{b}}) = \rho(\mathbf{A}_{\mathrm{f}})$. The right and left eigenvectors corresponding to the largest eigenvalue of $\mathbf{A}_{\mathrm{b}}$ can therefore be expressed as follows

$$\mathbf{r}_{\mathrm{b}}^T = \boldsymbol{l}_{\mathrm{f}}^T \quad \text{and} \quad \boldsymbol{l}_{\mathrm{b}} = \mathbf{r}_{\mathrm{f}}. \tag{A.10}$$

The state-distribution vector is of course the same as in the forward trellis, i.e.

$$\boldsymbol{\pi}_{\mathrm{b}} = \boldsymbol{\pi}_{\mathrm{f}}. \tag{A.11}$$

The optimal (backward) state-transition probabilities are obtained as follows

$$\mathbf{Q}_{\mathrm{b}}^*(i,j) = \begin{cases} \frac{r_{\mathrm{b}}(j)}{r_{\mathrm{b}}(i)} \frac{1}{\rho(\mathbf{A}_{\mathrm{b}})} = \frac{l_{\mathrm{f}}(j)}{l_{\mathrm{f}}(i)} \frac{1}{\rho(\mathbf{A}_{\mathrm{f}})} & \forall (i,j) : \mathbf{A}_{\mathrm{b}}(i,j) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{A.12}$$

and are collected in the (backward) state-transition probability matrix $\mathbf{Q}_{\mathrm{b}}^*$, i.e.

$$\mathbf{Q}_{\mathrm{b}}^* = \begin{pmatrix} 0.5437 & 0.2956 & 0.1607 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} . \tag{A.13}$$

As $\mathbf{Q}_\mathrm{b}$ is a stochastic matrix, its largest eigenvalue equals one. Both state-transition probability matrices, the forward and the backward, exhibit the same row constraint. The right and left eigenvectors corrsponding to the largest eigenvalue of the forward and the backward state-transition probability matrix are therefore identical, i.e.

$$\check{\boldsymbol{r}}_\mathrm{b}^T = \check{\boldsymbol{r}}_\mathrm{f}^T \quad \text{and} \quad \check{\boldsymbol{l}}_\mathrm{b} = \check{\boldsymbol{l}}_\mathrm{f} \, . \tag{A.14}$$

Note that in general, it does not hold that

$$\mathbf{Q}_\mathrm{b} = \mathbf{Q}_\mathrm{f}^T \tag{A.15}$$

even if

$$\mathbf{A}_\mathrm{b} = \mathbf{A}_\mathrm{f}^T. \tag{A.16}$$

Example: $(1,\infty)$-RLL source.

A.3 Parallel Branches

Now, we show that the extension of the results in previous sections to Markov sources with parallel branches is straightforward. We extend the notion of STPs to branch-transition probabilities (BTPs), i.e.

$$\mathbf{Q}_{||}(i,\ell,j) \triangleq Pr(S_k = j, B_k = \ell | S_{k-1} = i) \tag{A.17}$$

with the constraint

$$\sum_{j\in\mathsf{S}} \sum_{\left(\mathrm{lst}(\ell)=i\right)\cap\left(\mathrm{rst}(\ell)=j\right)} \mathbf{Q}_{||}(i,\ell,j) = \sum_{j\in\mathsf{S}} \mathbf{Q}(i,j) = 1 \quad \forall i \in \mathsf{S}. \tag{A.18}$$

The optimal BTPs are given as shown in [70] by

$$\mathbf{Q}_{||}^*(i,\ell,j) = \begin{cases} \frac{\boldsymbol{r}(j)}{\boldsymbol{r}(i)} \frac{\mathbf{A}_{||}(i,\ell,j)}{\rho(\mathbf{A})} & \forall (i,j) : \mathbf{A}(i,j) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{A.19}$$

where $\boldsymbol{r}^T$ is the right eigenvector belonging to the largest eigenvalue of $\mathbf{A}$ with $\mathbf{A}(i,j) = \sum_\ell \mathbf{A}_{||}(i,\ell,j)$. Consequently, we can write the entropy rate of the Markov process as

$$H(\mathcal{X}) = \sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i) \sum_{j\in\mathsf{S}} \sum_{\left(\mathrm{lst}(\ell)=i\right)\cap\left(\mathrm{rst}(\ell)=j\right)} \mathbf{Q}(i,\ell,j) \log_2 \frac{\mathbf{A}_{||}(i,\ell,j)}{\mathbf{Q}(i,\ell,j)}. \tag{A.20}$$

Example A.1 ($(0,2)$-RLL Source with Parallel Branches)
Take two consecutive trellis sections of the $(0,2)$-RLL source and shrink them to one single trellis section, i.e. we take the 2nd power of the adjaceny matrix $\mathbf{A}$. This gives

$$\mathbf{A}_{\mathrm{f}}^2 = \begin{pmatrix} 2 & 1 & 1 \\ 2 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

with $\rho(\mathbf{A}_{\mathrm{f}}^2) = 3.3830$. The optimal STPs are

$$\mathbf{Q}_{\mathrm{f}}^* = \begin{pmatrix} 0.4196 & 0.3522 & 0.2282 \\ 0.5437 & 0.4563 & 0 \\ 0.5437 & 0.4563 & 0 \end{pmatrix} .$$

The BTPs are $\mathbf{Q}_{||}(1,1,1) = \mathbf{Q}_{||}(1,2,1) = 0.4196/2.0$ and $\mathbf{Q}_{||}(2,1,1) = \mathbf{Q}_{||}(2,2,1) = 0.5437/2.0$ as $\mathbf{A}_{||}(1,1,1) = \mathbf{A}_{||}(1,2,1)$ and $\mathbf{A}_{||}(2,1,1) = \mathbf{A}_{||}(2,2,1)$ respectively. The state-distribution vector remains of course unchanged and is given by (A.4). Plugging these number into (A.20) delivers 1.7583 and equals the logarithm to the base two of $\rho(\mathbf{A}_{\mathrm{f}}^2)$. The entropy rate in bits/symbol is obtained by deviding by two, i.e. $C = 0.8792$ bits/symbol.

Thus, the results from Chapter 2 and 3 can be extended to parallel branches simply by replacing the STPs with BTPs. The a-posteriori state-transition weight matrix $\mathbf{T}$ generalizes in a staightforward manner to the a-posteriori *branch*-transition weight matrix.

Definition A.2 (A-posteriori Branch-Transition Weight) *For all branches (i,ℓ,j) : $\mathbf{A}_{||}(i,\ell,j) = 1$, we define the a-posteriori* branch-*transition matrix $\mathbf{T}_{||}$ with a-posteriori* branch-*transition weights as entries in the following way:*

$$\mathbf{T}_{||}(i,\ell,j) \triangleq \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} E_{p_{Y^n|S_{k-1}=i,B_k=\ell,S_k=j}} \left[\log_2 \frac{v_k(i,\ell,j|Y^n)}{v_{k-1}(i|Y^n)} \right] \tag{A.21}$$

$$= \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} E_{r_{Y^n}} \left[\log_2 \frac{v_k(i,\ell,j|Y^n)^{\frac{v_k(i,\ell,j|Y^n)}{\boldsymbol{\pi}(i)\mathbf{Q}(i,\ell,j)}}}{v_{k-1}(i|Y^n)^{\frac{v_{k-1}(i|Y^n)}{\boldsymbol{\pi}(i)}}} \right] \tag{A.22}$$

where we used the definitions

$$v_k(i,\ell,j|Y^n) \triangleq Pr(S_{k-1} = i, B_k = \ell, S_k = j|Y^n) \tag{A.23}$$

$$v_{k-1}(i|Y^n) \triangleq Pr(S_{k-1} = i|Y^n). \tag{A.24}$$

If $\mathbf{A}_{||}(i,\ell,j) = 0$, we define $\mathbf{T}_{||}(i,\ell,j) \triangleq -\infty$.

The noisy adjaceny matrix becomes then to

$$\tilde{\mathbf{A}}(i,j) = \begin{cases} \sum_\ell 2^{\mathbf{T}_{||}(i,\ell,j)} & \forall (i,\ell,j) : \mathbf{A}_{||}(i,\ell,j) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{A.25}$$

and the updated branch-transition probabilities are now given by

$$\mathbf{Q}_{||}(i,\ell,j) = \frac{\tilde{\boldsymbol{r}}(j)}{\tilde{\boldsymbol{r}}(i)} \cdot \frac{2^{\mathbf{T}_{||}(i,\ell,j)}}{\rho(\tilde{\mathbf{A}})} \quad , \tag{A.26}$$

where $\tilde{\boldsymbol{r}}^T$ is the right Perron-vector of $\tilde{\mathbf{A}}$.

Appendix B

The Perron-Eigenvalue: Theory and Practice

Definition B.1 (Non-negative Matrix) *A matrix* $\mathbf{A}$ *of size* $L \times L$ *is called non-negative if all its entries* $\mathbf{A}(i,j)$ *are non-negative, i.e. if* $\mathbf{A}(i,j) \geq 0$ for $i,j = \{1,2,\ldots,L\}$.

Definition B.2 (Positive Matrix) *A matrix* $\mathbf{A}$ *of size* $L \times L$ *is called positive if all its entries* $\mathbf{A}(i,j)$ *are positive, i.e. if* $\mathbf{A}(i,j) > 0$ $i,j = \{1,2,\ldots,L\}$.

Similar definitions hold for vectors. The entries of a positive row vector $\boldsymbol{v}$ of size $1 \times L$ are therefore all positive, i.e $\boldsymbol{v}(i) > 0$ for $i = \{1,2,\ldots,L\}$.

Definition B.3 (Spectral Radius) *The eigenvalue with the largest absolute value of the matrix* $\mathbf{A}$ *is termed spectral radius of* $\mathbf{A}$ *and denoted by* $\rho(\mathbf{A})$.

B.1 Perron-Frobenius Theorem

For positive matrices the Perron theorem holds [43]. The name Frobenius is associated with generalizations of Perron's results about positive matrices to non-negative matrices.

Theorem B.1 (Perron-Frobenius Theorem [43]) *Let* $\mathbf{A}$ *be an irreducible and non-negative matrix. Then*

a.) $\rho(\mathbf{A}) > 0$.

b.) $\rho(\mathbf{A})$ *is an eigenvalue of* $\mathbf{A}$.

c.) There are positive right and left eigenvectors, $\boldsymbol{r}^T$ *and* $\boldsymbol{l}$*, such that* $\mathbf{A}\boldsymbol{r}^T = \rho(\mathbf{A})\boldsymbol{r}^T$ *and* $\boldsymbol{l}\mathbf{A} = \rho(\mathbf{A})\boldsymbol{l}$*, respectively.*

d.) $\rho(\mathbf{A})$ is an algebraically (and hence geometrically) simple eigenvalue of $\mathbf{A}$.

For a proof see [43]. This so-called *Perron-Frobenius theorem* guarantees that the eigenspace of an irreducible and non-negative matrix associated with the *Perron-root* (eigenvalue with largest absolute value) is one-dimensional (unique). The unique positive left/right eigenvector associated with the Perron-root whose components sum up to 1 is called the left/right *Perron-vector*.

Corollary B.2 *Suppose $\mathbf{A}$ is irreducible and non-negative and suppose the set $\Lambda = \{\lambda_1 = \rho(\mathbf{A}), \lambda_2, \ldots, \lambda_{1+(k-1)}\}$ of eigenvalues of maximum modulus (absolute value) has exactly k distinct elements. Then each eigenvalue $\lambda_i \in \Lambda$ has algebraic multiplicity 1, i.e. is unique, and*

$$\Lambda = \{e^{2\pi i\ell/k}\rho(\mathbf{A}) : \ell = 0, 1, \ldots, k-1\} \tag{B.1}$$

that is, these maximum modulus eigenvalues are precisely the k-th roots of unity times $\rho(\mathbf{A})$. Moreover, if λ is any *eigenvalue of $\mathbf{A}$, then $e^{2\pi i\ell/k}\lambda$ is an eigenvalue for all $\ell = 0, 1, \ldots, k-1$.*

Non-negative matrices may therefore have several eigenvalues with the same modulus (absolute value). For positive matrices, this is different. There is only *one* eigenvalue with maximum modulus [43]. For the power method (see next subsection), we need exactly this property.

Definition B.4 (Primitive Matrix) *A non-negative matrix $\mathbf{A}$ is said to be primitive if it is irreducible and has only one eigenvalue of maximum modulus.*

We restrict ourselves therefore to non-negative matrices that are *primitive*. Moreover, the following theorem holds for non-negative matrices [43].

Theorem B.3 (Mixing Property) *If $\mathbf{A}$ is non-negative, then $\mathbf{A}$ is primitive if and only if $\mathbf{A}^n > \mathbf{0}$ for some $n \geq 1$.*

We recognize that this theorem is equivalent to the *mixing property* for irreducible and aperiodic FSM (see Chapter 2). Thus, the n-th power of a non-negative primitive matrix becomes a positive matrix and then the Perron theorem applies which guarantees that there is only one eigenvalue of maximum modulus.

B.2 Power-Method

The *power-method* [63] is a very practical method for computing in a straightforward way one eigenvalue λ (usually the largest in magnitude) and corresponding left eigenvector $\boldsymbol{l}$ of a non-negative primitive matrix $\mathbf{A}$. It is an iterative method, in which we start with an initial guess $\hat{\boldsymbol{l}}^0$ of $\boldsymbol{l}$ and generate a sequence of approximations $\hat{\boldsymbol{l}}^k$ that converges under certain conditions to $\boldsymbol{l}$ as $k \to \infty$.

Algorithm B.1 Power-Method

Init: *Choose an initial guess* $\hat{\boldsymbol{l}}^{(0)}$*, fix an accuracy goal* δ*, and set* $g_0 = 1$.

Loop: ***Step 0:*** *Compute* $\boldsymbol{\gamma}^{(k+1)} = \hat{\boldsymbol{l}}^{(k)}\mathbf{A}$.

Step 1: *Select the largest (in absolute value) element of* $\boldsymbol{\gamma}^{(k+1)}$*, i.e.* $g_{k+1} = \max_i(|\boldsymbol{\gamma}^{(k+1)}(i)|)$.

Step 2: *Scale* $\hat{\boldsymbol{l}}^{(k+1)} = \boldsymbol{\gamma}^{(k+1)}/g_{k+1}$.

Step 3: `IF` $|g_{k+1} - g_k| < \delta$ `STOP`.
`ELSE` *increment* k *by one and* `GOTO` *Step 0.*

B.2.1 Convergence Criteria

Suppose that the $L \times L$ matrix $\mathbf{A}$ has a complete set of left eigenvectors $\boldsymbol{l}_1, \ldots, \boldsymbol{l}_L$ with corresponding eigenvalues $\lambda_1, \ldots, \lambda_L$. Without loss of generality, we can arrange the eigenvalues in decreasing order, that is

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_L|. \tag{B.2}$$

Moreover, we scale the eigenvectors $\boldsymbol{l}_i$ such that the absolute largest element of each is equal to 1 and write the initial guess $\hat{\boldsymbol{l}}^{(0)}$ as a linear combination of the eigenvectors of $\mathbf{A}$, i.e.

$$\hat{\boldsymbol{l}}^{(0)} = \sum_{i=1}^{L} c_i \boldsymbol{l}_i. \tag{B.3}$$

Applying a single step of the power-method to this vector yields

$$\boldsymbol{\gamma}^{(1)} = \hat{\boldsymbol{l}}^{(0)} \cdot \mathbf{A} = \sum_{i=1}^{L} \lambda_i c_i \boldsymbol{l}_i \tag{B.4}$$

and we obtain after the first iteration

$$\hat{\boldsymbol{l}}^{(1)} = \frac{\boldsymbol{\gamma}^{(1)}}{g_1} = \frac{1}{g_1} \sum_{i=1}^{L} \lambda_i c_i \boldsymbol{l}_i. \tag{B.5}$$

After k iterations, we have

$$\hat{\boldsymbol{l}}^{(k)} = \frac{1}{g_1 g_2 \cdots g_k} \sum_{i=1}^{n} \lambda_i^k c_i \boldsymbol{l}_i \tag{B.6}$$

$$= \frac{c_1 \lambda_1^k}{g_1 g_2 \cdots g_k} \left[\boldsymbol{l}_1 + \sum_{i=2}^{L} \left(\frac{\lambda_i}{\lambda_1} \right)^k \frac{c_i}{c_1} \boldsymbol{l}_i \right]. \tag{B.7}$$

Now the idea is that since λ_1 is the largest (in absolute value) eigenvalue, all the terms apart from the first in the expression inside the braces tend to zero as $k \to \infty$. Thus, the eigenvector $\boldsymbol{l}_1$ dominates the behavior of $\hat{\boldsymbol{l}}^{(k)}$ for large k. This will fail if there is another eigenvalue of the same absolute value as λ_1. Therefore, we assume that λ_1 is bigger than any other eigenvalue of $\mathbf{A}$, i.e.

$$|\lambda_1| > |\lambda_i| \quad \forall\ i = 2, 3, \ldots, L. \tag{B.8}$$

This assumption is fulfilled if the matrix $\mathbf{A}$ is primitive (see previous subsection). Note, we use here absolute values as the matrix $\mathbf{A}$ may have complex entries. With property B.8, we can assert that

$$\left(\frac{\lambda_i}{\lambda_1}\right)^k \to 0 \quad \text{as } k \to \infty, \tag{B.9}$$

for all $i > 1$ and thus

$$\hat{\boldsymbol{l}}^{(k)} \to \frac{c_1 \lambda_1^k}{g_1 g_2 \cdots g_k} \boldsymbol{l}_1 \quad \text{as } k \to \infty. \tag{B.10}$$

We know that $\hat{\boldsymbol{l}}^{(k)}$ has the property that the largest (in absolute value) element is always equal to one, as does $\boldsymbol{l}_1$. Therefore the constant of proportionality between $\hat{\boldsymbol{l}}^{(k)}$ and $\boldsymbol{l}_1$ must approach unity for large k, that is

$$\hat{\boldsymbol{l}}^{(k)} \to \boldsymbol{l}_1 \quad \text{as } k \to \infty, \tag{B.11}$$

and

$$\frac{c_1 \lambda_1^k}{g_1 g_2 \cdots g_k} \to 1 \quad \text{as } k \to \infty. \tag{B.12}$$

This can only occur if

$$g_k \to \lambda_1 \quad \text{as } k \to \infty. \tag{B.13}$$

Hence, the scaling factor g_k will converge to λ_1, the largest eigenvalue of $\mathbf{A}$, for k large. In practice, one would continue iterating up to a prescribed accuracy goal, i.e. till $|g_{k+1} - g_k| < \delta$.

B.2.2 Remarks

The convergence rate depends mainly on the mixing property of $\mathbf{A}$. It can be expressed as the ratio of the largest to the second largest eigenvalue, i.e.

$$\frac{|\lambda_2|}{|\lambda_1|}, \tag{B.14}$$

which is termed *spectral gap*.

Instead of using the largest entry of $\boldsymbol{\gamma}^{(k+1)}$ as scaling factor, it is also possible to use other scaling factors; e.g. g_{k+1} can be a normalizing factor such that the sum of the entries of $\hat{\boldsymbol{l}}^{(k+1)}$ always equals one, i.e.

$$g_{k+1} = \sum_{i=1}^{L} \boldsymbol{\gamma}^{(k+1)}(i) \tag{B.15}$$

(assuming that all entries are reals).

B.2.3 Connection to the Sum-Product Algorithm

Any vector-matrix multiplication, requires products and sums of the vector and matrix entries. For the computation of the left Perron-vector one iteration of the power-method described above involves the computation of the vector-matrix multiplication with the estimate $\hat{\boldsymbol{l}}^{(k)}$ and $\mathbf{A}$. For a two-by-two matrix $\mathbf{A}$, this multiplication step yields

$$\begin{aligned}
\boldsymbol{\gamma}^{(k+1)} &= \hat{\boldsymbol{l}}^{(k)} \cdot \mathbf{A} && \text{(B.16)}\\
&= \begin{pmatrix} \boldsymbol{l}^{(k)}(1) & \boldsymbol{l}^{(k)}(2) \end{pmatrix} \cdot \begin{pmatrix} \mathbf{A}(1,1) & \mathbf{A}(1,2) \\ \mathbf{A}(2,1) & \mathbf{A}(2,2) \end{pmatrix} && \text{(B.17)}\\
&= \begin{pmatrix} \boldsymbol{l}^{(k)}(1)\mathbf{A}(1,1) + \boldsymbol{l}^{(k)}(2)\mathbf{A}(2,1) \\ \boldsymbol{l}^{(k)}(1)\mathbf{A}(1,2) + \boldsymbol{l}^{(k)}(2)\mathbf{A}(2,2) \end{pmatrix}^{T}. && \text{(B.18)}
\end{aligned}$$

One such iteration step of the power-method can be illustrated by a single trellis section of the DICODE channel (see Fig. 2.2, right part). It equals one step of the forward (left-to-right) computation of the sum-product algorithm with $\mathbf{A}$ containing the branch metrics and $\hat{\boldsymbol{l}}^{(k)}$ the old (left) state metrics and $\boldsymbol{\gamma}^{(k+1)}$ the new (right) state metrics that are not scaled. The purpose of the scaling factor g_{k+1} is the same as of φ_{k+1} in the forward recursion of the sum-product algorithm: i.e. to prevent that the entries of $\boldsymbol{\gamma}^{(k+1)}$ become unbounded. Of course, the reverse operation, backward or right-to-left recursion of the sum-product algorithm, is also possible and delivers by the same reasoning the right Perron-vector.

Hence, the computation of the left (right) Perron-vector and the largest eigenvalue of $\mathbf{A}$ is performed by a forward or left-to-right (backward or right-to-left) recursion of the sum-product algorithm operating on the trellis induced by $\mathbf{A}$.

Appendix C

Derivation of Capacity-Achieving State-Transition Probabilities

The optimal (maxentropic) state-transition probabilities $\mathbf{Q}^*(i,j)$ of a Markov source result from maximizing the entropy rate

$$H(\mathcal{X}) = -\sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i) \sum_{j\in\mathsf{S}} \mathbf{Q}(i,j) \left[\log_2 \mathbf{Q}(i,j) - \mathbf{T}(i,j)\right] \tag{C.1}$$

under the following constraints for all transitions $(i,j) : \mathbf{A} = 1$

$$\mathbf{Q}(i,j) \geq 0 \tag{C.2}$$

$$\sum_{j\in\mathsf{S}} \mathbf{Q}(i,j) = 1 \quad \forall\ i \in \mathsf{S} \tag{C.3}$$

$$\sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i)\mathbf{Q}(i,j) = \boldsymbol{\pi}(j) \quad \forall\ j \in \mathsf{S} \tag{C.4}$$

$$\sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i) = 1 \tag{C.5}$$

where $\mathbf{T}(i,j)$, for the time being, can be any real number associated with the transition from state i to state j. For all transitions $(i,j) : \mathbf{A} = 0$, it holds that $\mathbf{Q}(i,j) = 0$.

We set up now the Lagrangian for this constraint optimization problem without the first constraint, i.e. the constraint that the state-transition probabilities must be non-negative.

Later, we will show that indeed this constraint is met. The Lagrangian is given as follows

$$\begin{aligned}\mathcal{L} = &- \sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i) \sum_{j\in\mathsf{S}} \mathbf{Q}(i,j) \left[\log_2 \mathbf{Q}(i,j) - \mathbf{T}(i,j)\right] \\ &+ \sum_{i\in\mathsf{S}} \lambda_i \sum_{j\in\mathsf{S}} \mathbf{Q}(i,j) + \sum_{j\in\mathsf{S}} \lambda_j' \left(\sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i)\mathbf{Q}(i,j) - \boldsymbol{\pi}(j) \right) \\ &+ \lambda'' \sum_{i\in\mathsf{S}} \boldsymbol{\pi}(i). \end{aligned} \tag{C.6}$$

We have to solve the equations

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\pi}(k)} \stackrel{!}{=} 0 \quad \forall k \in \mathsf{S} \tag{C.7}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Q}(k,\ell)} \stackrel{!}{=} 0 \quad \forall k,\ell \in \mathsf{S} \tag{C.8}$$

where $\mathcal{L}$ is a concave function in Q defined on a convex domain (for a proof refer to the last subsection in this appendix).

Taking the derivative of $\mathcal{L}$ with respect to $\boldsymbol{\pi}(k)$ and $\mathbf{Q}(k,\ell)$ gives

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\pi}(k)} = -\sum_{j\in\mathsf{S}} \mathbf{Q}(k,j) \left[\log_2 \mathbf{Q}(k,j) - \mathbf{T}(k,j)\right] + \sum_{j\in\mathsf{S}} \lambda_j' \mathbf{Q}(k,j) - \lambda_k' + \lambda'' \stackrel{!}{=} 0 \tag{C.9}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Q}(k,\ell)} = -\boldsymbol{\pi}(k) \left[\log_2 \mathbf{Q}(k,\ell) + \frac{\mathbf{Q}(k,\ell)}{\mathbf{Q}(k,\ell)}\right] + \boldsymbol{\pi}(k)\mathbf{T}(k,\ell) + \lambda_k + \lambda_\ell' \boldsymbol{\pi}(k) \stackrel{!}{=} 0. \tag{C.10}$$

Multiplying the second equation by $\mathbf{Q}(k,\ell)$ and summing over $\ell \in \mathsf{S}$, we obtain by means of the second constraint (C.3)

$$-\boldsymbol{\pi}(k) \sum_{\ell\in\mathsf{S}} \mathbf{Q}(k,\ell) \left[\log_2 \mathbf{Q}(k,\ell) - \mathbf{T}(k,\ell)\right] - \boldsymbol{\pi}(k) + \lambda_k + \boldsymbol{\pi}(k) \sum_{\ell\in\mathsf{S}} \lambda_\ell' \mathbf{Q}(k,\ell) = 0. \tag{C.11}$$

We can bring the first equation to a similar form by replacing the summations over j by summations over ℓ and multiplying by $\boldsymbol{\pi}(k)$. This yields

$$-\boldsymbol{\pi}(k) \left[\sum_{\ell\in\mathsf{S}} \mathbf{Q}(k,\ell) \left[\log_2 \mathbf{Q}(k,\ell) - \mathbf{T}(k,\ell)\right] - \sum_{\ell\in\mathsf{S}} \lambda_\ell' \mathbf{Q}(k,\ell) + \lambda_k' - \lambda'' \right] = 0 \tag{C.12}$$

Subtracting (C.12) from (C.11) delivers

$$-\boldsymbol{\pi}(k) + \lambda_k + \boldsymbol{\pi}(k)\lambda_k' - \boldsymbol{\pi}(k)\lambda'' = 0 \tag{C.13}$$

and finally

$$\lambda_k = (1 - \lambda_k' + \lambda'') \cdot \boldsymbol{\pi}(k). \tag{C.14}$$

Replacing this λ_k in (C.10) yields

$$\boldsymbol{\pi}(k) \cdot \left(-\log_2 \mathbf{Q}(k,\ell) + \mathbf{T}(k,\ell) - \lambda_k' + \lambda'' + \lambda_\ell' \right) = 0. \tag{C.15}$$

Because the underlying finite-state Markov model is irreducible, $\boldsymbol{\pi}(k) > 0$ for all $k \in \mathsf{S}$. Thus it must hold that

$$\mathbf{Q}(k,\ell) = \exp\left(\lambda_\ell^{'} - \lambda_k^{'} + \lambda^{''} + \mathbf{T}(k,\ell)\right) \quad \forall k,\ell \in \mathsf{S} \tag{C.16}$$

and we note that the first constraint (C.2) is indeed fulfilled. Assume that the matrix $\mathbf{A}$ consists of the entries $\mathbf{A}(k,\ell) = e^{\mathbf{T}(k,\ell)}$ if the state-transition exists. Moreover, let the vector $\boldsymbol{r}$ be the vector with entries $\boldsymbol{r}(k) = e^{\lambda_k^{'}}$ and $\rho = e^{-\lambda^{''}}$. Inserting these results into (C.16), we get

$$\mathbf{Q}(k,\ell) = \frac{\boldsymbol{r}(\ell)}{\boldsymbol{r}(k)} \cdot \frac{\mathbf{A}(k,\ell)}{\rho}. \tag{C.17}$$

From the global balance condition (C.4)[1] we obtain

$$\sum_{k\in\mathsf{S}} \boldsymbol{\pi}(k) \frac{\boldsymbol{r}(\ell)}{\boldsymbol{r}(k)} \cdot \frac{\mathbf{A}(k,\ell)}{\rho} = \boldsymbol{\pi}(\ell) \quad \forall \ell \in \mathsf{S} \tag{C.18}$$

and by letting the vector $\boldsymbol{l}$ have entries $\boldsymbol{l}(k) = \boldsymbol{\pi}(k)/\left(c \cdot \boldsymbol{r}(k)\right)$ we can rewrite (C.18) as follows

$$\boldsymbol{l} \cdot \mathbf{A} = \rho \cdot \boldsymbol{l}, \tag{C.19}$$

i.e. $\boldsymbol{l}$ is a left eigenvector of $\mathbf{A}$ with eigenvalue ρ, whose entries must be non-negative. Consequently, to fulfill the third constraint in (C.5), we must have

$$\boldsymbol{\pi}(i) = c \cdot \boldsymbol{l}(i) \cdot \boldsymbol{r}(i) \quad \forall i \in \mathsf{S} \qquad \text{with } c = \frac{1}{\sum_{i\in\mathsf{S}} \boldsymbol{l}(i)\boldsymbol{r}(i)}. \tag{C.20}$$

The maximal value of $H(\mathcal{X})$ becomes now to

$$H(\mathcal{X}) = -\sum_{i\in\mathsf{S}} c \cdot \boldsymbol{l}(i) \cdot \boldsymbol{r}(i) \sum_{j\in\mathsf{S}} \left(\frac{\boldsymbol{r}(j)}{\boldsymbol{r}(i)} \cdot \frac{\mathbf{A}(i,j)}{\rho} \cdot \left[\log_2 \left(\frac{\boldsymbol{r}(j)}{\boldsymbol{r}(i)} \cdot \frac{\mathbf{A}(i,j)}{\rho} \right) - \mathbf{T}(i,j) \right] \right). \tag{C.21}$$

Noting that $\mathbf{A}(i,j) = e^{\mathbf{T}(i,j)}$, we obtain

$$\begin{aligned} H(\mathcal{X}) = &\underbrace{-\frac{c}{\rho} \sum_{i\in\mathsf{S}} \boldsymbol{l}(i) \sum_{j\in\mathsf{S}} \boldsymbol{r}(j) \cdot \mathbf{A}(i,j) \log_2 \boldsymbol{r}(j)}_{c\sum_j \boldsymbol{\pi}(j)\log_2 \boldsymbol{r}(j)} \\ &\underbrace{+\frac{c}{\rho} \sum_{i\in\mathsf{S}} \boldsymbol{l}(i) \sum_{j\in\mathsf{S}} \boldsymbol{r}(j) \cdot \mathbf{A}(i,j) \log_2 \boldsymbol{r}(i)}_{c\sum_i \boldsymbol{\pi}(i)\log_2 \boldsymbol{r}(i)} \\ &\underbrace{+\frac{c}{\rho} \sum_{i\in\mathsf{S}} \boldsymbol{l}(i) \sum_{j\in\mathsf{S}} \boldsymbol{r}(j) \cdot \mathbf{A}(i,j) \log_2 \rho}_{\log_2 \rho} \\ = &\log_2 \rho. \end{aligned} \tag{C.22}$$

[1] Alternatively we could also continue with (C.3).

Thus the entropy rate is maximized for ρ being the largest eigenvalue of $\mathbf{A}$, i.e. $\rho = \rho(\mathbf{A})$. This maximization is only possible if the conditions that the right eigenvector corresponding to the eigenvalue $\rho(\mathbf{A})$ has positive entries and the left eigenvector has non-negative entries. For an irreducible and non-negative matrix $\mathbf{A}$, one can indeed show that these conditions can be met [43]. The Perron-Forbenius theorem guarantees this. Hence, the maxentropic $\mathbf{Q}^*(i,j)$ are given by (C.17) with ρ being the eigenvalue with maximum modulus (absolute value) of $\mathbf{A}$. The maximal entropy rate equals the logarithm of $\rho(\mathbf{A})$.

Finally, the 2nd derivative must be negative such that the optimized solution is indeed a maximum. From (C.9) and (C.10) it follows that this is indeed the case. Thus, the maximal entropy rate is given by (C.22) and the optimal STPs are given by (C.17).

Concavity of the Entropy Rate of Markov Sources

We provide a proof that the entropy rate of a Markov process is a concave function over the manifold $\overline{\mathsf{Q}}$.

Assume that $\overline{\mathsf{Q}}_1$ and $\overline{\mathsf{Q}}_2$ are two valid sets of joint state-transitions probabilities, i.e. the Markov properties hold for all entries of $\overline{\mathbf{Q}}_1$ and $\overline{\mathbf{Q}}_2$. Thus, for all transitions $(i,j) : \mathbf{A}(i,j) = 1$ it must hold that

$$\overline{\mathbf{Q}}_1(i,j) \geq 0 \text{ and } \overline{\mathbf{Q}}_2(i,j) \geq 0 \tag{C.23}$$

$$\sum_{i\in\mathsf{S}}\sum_{j\in\mathsf{S}} \overline{\mathbf{Q}}_1(i,j) = 1 \text{ and } \sum_{i\in\mathsf{S}}\sum_{j\in\mathsf{S}} \overline{\mathbf{Q}}_2(i,j) = 1 \tag{C.24}$$

$$\sum_{i\in\mathsf{S}} \overline{\mathbf{Q}}_1(i,j) = \sum_{k\in\mathsf{S}} \overline{\mathbf{Q}}_1(j,k) \text{ and } \sum_{i\in\mathsf{S}} \overline{\mathbf{Q}}_2(i,j) = \sum_{k\in\mathsf{S}} \overline{\mathbf{Q}}_2(j,k). \tag{C.25}$$

The corresponding state-probabilities are

$$\boldsymbol{\pi}_1(i) = \sum_{j\in\mathsf{S}} \overline{\mathbf{Q}}_1(i,j) \text{ and } \boldsymbol{\pi}_2(i) = \sum_{j\in\mathsf{S}} \overline{\mathbf{Q}}_2(i,j). \tag{C.26}$$

We form new joint state-transition probabilities in the following way

$$\overline{\mathbf{Q}}_3(i,j) = \alpha \cdot \overline{\mathbf{Q}}_1(i,j) + (1-\alpha) \cdot \overline{\mathbf{Q}}_2(i,j) \tag{C.27}$$

for $0 \leq \alpha \leq 1$. As $\overline{\mathsf{Q}}_1$ and $\overline{\mathsf{Q}}_2$ fulfill the Markov constraints, $\overline{\mathsf{Q}}_3$ does fulfill them as well. The new state-probabilities are then given by

$$\boldsymbol{\pi}_3(i) = \sum_{j\in\mathsf{S}} \overline{\mathbf{Q}}_3(i,j) \tag{C.28}$$

$$= \alpha \sum_{j\in\mathsf{S}} \overline{\mathbf{Q}}_1(i,j) + (1-\alpha) \sum_{j\in\mathsf{S}} \overline{\mathbf{Q}}_2(i,j) \tag{C.29}$$

$$= \alpha\boldsymbol{\pi}_1(i) + (1-\alpha)\boldsymbol{\pi}_2(i). \tag{C.30}$$

With these prerequisites and assuming that $\mathbf{M}(i,j)$ is an entry of a matrix $\mathbf{M}$ that does not depend on $\overline{\mathsf{Q}}_1$ nor on $\overline{\mathsf{Q}}_2$, we obtain the following concavity relation for the entropy rate $H(\mathcal{X};\overline{\mathsf{Q}}_3)$ of a Markov process $\mathcal{X}$ parameterized by the manifold $\overline{\mathsf{Q}}_3$:

$$H(\mathcal{X};\overline{\mathsf{Q}}_3) = \sum_{(i,j):\mathbf{A}(i,j)=1} \overline{\mathbf{Q}}_3(i,j)\log_2 \frac{\boldsymbol{\pi}_3(i)\mathbf{M}(i,j)}{\overline{\mathbf{Q}}_3(i,j)} \tag{C.31}$$

$$= \sum_{(i,j):\mathbf{A}(i,j)=1} \Bigg[\big[\alpha\cdot\overline{\mathbf{Q}}_1(i,j)+(1-\alpha)\cdot\overline{\mathbf{Q}}_2(i,j)\big] \log_2 \frac{\big(\alpha\boldsymbol{\pi}_1(i)+(1-\alpha)\boldsymbol{\pi}_2(i)\big)\,\mathbf{M}(i,j)}{\alpha\cdot\overline{\mathbf{Q}}_1(i,j)+(1-\alpha)\cdot\overline{\mathbf{Q}}_2(i,j)}\Bigg] \tag{C.32}$$

$$\geq \sum_{(i,j):\mathbf{A}(i,j)=1} \Bigg[\alpha\overline{\mathbf{Q}}_1(i,j)\log_2 \frac{\alpha\boldsymbol{\pi}_1(i)\mathbf{M}(i,j)}{\alpha\overline{\mathbf{Q}}_1(i,j)} + (1-\alpha)\overline{\mathbf{Q}}_2(i,j)\log_2 \frac{(1-\alpha)\boldsymbol{\pi}_2(i)\mathbf{M}(i,j)}{(1-\alpha)\overline{\mathbf{Q}}_2(i,j)}\Bigg]$$

$$= \alpha H(\mathcal{X};\overline{\mathsf{Q}}_1) + (1-\alpha)H(\mathcal{X};\overline{\mathsf{Q}}_2) \tag{C.33}$$

where the inequality follows from the log-sum inequality [24].

Abbreviations

Signal Processing

AWGN	additive white Gaussian noise
BTP	branch-transition probability
dB	decibel
DC	frequency zero (digital current)
FIR	finite impulse response
FSM	finite-state model
HMM	hidden Markov model
HMP	hidden Markov process
ISI	intersymbol interference
SNR	signal-to-noise ratio
STP	state-transition probability

Information Theory

AEP	asymptotic equipartition property
BSC	binary symmetric channel
BPSK	binary phase-shift keying
CSLLB	conjectured Shamai-Laroia lower bound
DMC	discrete memoryless channel
FSC	finite-state channel
i.i.d.	independent and identically distributed
i.u.d.	independent and uniformly distributed
pdf	probability density function
pmf	probability mass function
w. p. 1	with probability 1

Magnetic Recording

EPR4	extended PR4
E^2PR4	extended EPR4
GMR	giant magnetoresistance
MF	medium noise factor
MNP	medium noise power
MR	magnetoresistance
MS	Markov source
NLTS	nonlinear transition shift
NPRML	noise predictive PRML
PR4	partial-response class-4 polynomial
PRML	partial-response maximum-likelihood
PSE	partial signal erasure
RLL	run-length limited
TRP	total receiver power (signal and medium noise)

List of Symbols

General

X	random variable
x	realization of X
X	alphabet of x
X^n	sequence of random variables, i.e. $X^n = (X_1, X_2, \ldots, X_n)$
$\mathcal{X}$	process
$\hat{s}$	estimate of a scalar s
$Pr(\cdot)$	probability
$\boldsymbol{v}$	row vector
$\mathbf{M}$	matrix
λ_i	eigenvalue of the $L \times L$ matrix $\mathbf{A}$ with $1 \leq i \leq L$; the eigenvalues are ordered, i.e. $\lvert\lambda_1\rvert \geq \lvert\lambda_2\rvert \geq \ldots \geq \lvert\lambda_L\rvert$.
$\rho(\mathbf{A})$	spectral radius of $\mathbf{A}$; largest (in absolute value) eigenvalue of $\mathbf{A}$, i.e. $\rho(\mathbf{A}) = \max_i \lvert\lambda_i\rvert = \lvert\lambda_1\rvert$.

Finite-State Models

S_k	time-k state
B_k	time-k branch
V_k	noiseless time-k output
Y_k	received time-k signal (noisy version of V_k)
Υ	dimension of the output space
O_k	observed time-k signal
L	number of states
$\boldsymbol{\theta}$	parameter vector
m	memory of the FSM

M memory of the new trellis used for the upper bound or the original channel
$\mathbf{A}$ (forward) adjacency matrix
$\boldsymbol{r}^T$ right Perron-vector of $\mathbf{A}$
$\boldsymbol{l}$ left Perron-vector of $\mathbf{A}$
$\mathbf{Q}$ (forward) state-transition probability matrix
$\boldsymbol{\pi}$ state-distribution vector
$\tilde{\mathbf{A}}$ noisy adjacency matrix
$\tilde{\boldsymbol{r}}^T$ right Perron-vector of $\tilde{\mathbf{A}}$
$\tilde{\boldsymbol{l}}$ left Perron-vector of $\tilde{\mathbf{A}}$
$\mathbf{T}$ matrix consisting of a-posteriori state-transition weights
$\mathbf{t}$ matrix consisting of skewed a-posteriori state-transition weights
η extension degree of the Markov source

Magentic Recording

ζ position along the track
$m_0(\zeta)$ average magentization profile along ζ
ν rotation speed of the disk
δ medium thickness
d head-to-surface distance
H_c coercivity
H_m magnetization strength
M_r remanent state
a transition width parameter
T bit spacing parameter
N number of microtracks
L_th erasure threshold
PW50 pulse width of the Lorentzian pulse at 50% amplitude
N_0 one-sided power spectral density
D discrete-time delay element
$g(t)$ Lorentizan pulse, step response
$h(t)$ dipulse, impulse response, $h(t) = \frac{1}{2}\big(g(t) - g(t-T)\big)$
J jitter random variable

Information Theory

$H(X)$	entropy
$h(X)$	differential entropy
$I(X;Y)$	mutual information between X and Y
$H(\mathcal{X})$	entropy rate
$I(\mathcal{X};\mathcal{Y})$	information rate
$D\big(p(\cdot)\|\|q(\cdot)\big)$	divergence between $p_X(\cdot)$ and $q_X(\cdot)$
C	capacity
C^{FSC}	FSC capacity
$I_\eta(\mathsf{Q})$	FSM information rate for the η extension of the Markov source determined by Q
C_η^{FSM}	FSM capacity for the η extension of the Markov source
n	trellis length
$q(\cdot)$	distribution of the input X
$w\left(\,\cdot\,\middle|\,\cdot\,\right)$	forward channel law of the original channel
$r(\cdot)$	distribution of the channel output O of the original channel
$v\left(\,\cdot\,\middle|\,\cdot\,\right)$	backward channel law of the original channel
$w_{\mathrm{a}}\left(\,\cdot\,\middle|\,\cdot\,\right)$	forward channel law of the auxiliary channel
$r_{\mathrm{a}}(\cdot)$	distribution of the output Y of the auxiliary-channel
$v_{\mathrm{a}}\left(\,\cdot\,\middle|\,\cdot\,\right)$	backward channel law of the auxiliary-channel
τ	threshold for the adaptive reduced-state lower bound
$\mu_{\mathrm{f}}(x)$	metric of x in forward direction
$\mu_{\mathrm{r}}(x)$	metric of x in backward (reverse) direction
φ_k	time-k forward scaling factor
β_k	time-k backward scaling factor

Constants

$\varepsilon, \epsilon, \psi$	constants
c	multi-purpose constant

Bibliography

[1] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Trans. Inform. Theory*, vol. 18, no. 1, pp. 14–20, July 1972.

[2] D. Arnold and E. Eleftheriou, "Computing information rates of magnetic recording channels in the presence of media noise," in *Proc. of the IEEE GLOBECOM 2002, Nov. 17 – 21, Taipei, Taiwan*, Nov. 2002.

[3] ——, "On the information-theoretic capacity of magnetic recording systems in the presence of medium noise," *IEEE Trans. Magnetics*, vol. 38, no. 5, pp. 2319–2321, Sept. 2002.

[4] D. Arnold, A. Kavčić, R. Kötter, H.-A. Loeliger, and P. O. Vontobel, "The binary jitter channel: a new model for magnetic recording," in *Proc. of the IEEE Intern. Symp. on Inform. Theory, June 25 – 30, Sorrento, Italy*, 2000, p. 433.

[5] D. Arnold and H.-A. Loeliger, "On the information rate of binary-input channels with memory," in *Proc. of the IEEE Intern. Conf. on Communications, June 11 – 14, Helsinki, Finland*, vol. 9, 2001, pp. 2692–2695.

[6] R. Ash, *Information Theory*. John Wiley & Sons, 1965.

[7] C. P. M. J. Baggen, "An information theoretic approach to timing jitter," PhD thesis, University of California at San Diego, USA, 1993.

[8] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[9] A. Barron, "The strong ergodic theorem for densities: generalized Shannon-McMillan-Breiman theorem," *Annals of Prob.*, vol. 13, no. 4, pp. 1292–1303, 1995.

[10] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology," Bull. Amer. Math. Soc., vol. 73, pp. 360–363, 1966.

[11] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," The Annals of Mathematical Statistics, vol. 37, pp. 1554–1563, 1966.

[12] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," The Annals of Mathematical Statistics, vol. 41, no. 1, pp. 164–171, 1970.

[13] L. E. Baum and G. R. Sell, "Growth transformation for functions on manifolds," Pacific Journal of Mathematics, vol. 27, no. 2, pp. 211–227, Jan. 1968.

[14] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, New York, 1993.

[15] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen, "The infinite hidden Markov model," *Advances in Neural Information Processing Systems*, vol. 14, pp. 577–585, 2002.

[16] H. N. Bertram, *The Theory of Magnetic Recording*. Cambridge University Press, Apr. 1994.

[17] P. Billingsley, *Probability and Measure*. John Wiley & Sons, New York, 1995.

[18] R. E. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inform. Theory*, vol. 18, no. 4, pp. 460–473, July 1972.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Preprint, 2003, available under `http://www.stanford.edu/~boyd/cvxbook.html`, as of 03-17-2003.

[20] J. P. Caroselli, "Modeling, analysis, and mitigation of medium noise in thin film magnetic recording channels," PhD thesis, University of California at San Diego, 1998.

[21] X. Chen, "Limit theorems for functionals of ergodic Markov chains with general state space," *Memoires of the AMS*, vol. 139, May 1999.

[22] P. R. Chevillat, E. Eleftheriou, and D. Maiwald, "Noise-predictive partial-response equalizers and applications," in *Proc. of the IEEE Intern. Conf. on Communications, June 14 – 17, Chicago, IL, USA*, vol. 2, June 1992, pp. 942–947.

[23] R. D. Cideciyan, F. Dolivo, R. Hermann, W. Hirt, and W. Scott, "A PRML system for digital magnetic recording," *IEEE J. Sel. Areas Comm.*, vol. 10, pp. 38–56, Jan. 1992.

[24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[25] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Budapest: Akadémiai Kiadó (Publishing house of the Hungarian Academy of Sciences), 1981.

[26] A. Dholakia, E. Eleftheriou, and T. Mittelholzer, "On iterative decoding for magnetic recording channels," in *Proc. of the 2nd Int. Symp. on Turbo Codes & Related Topics, Brest, France, September 4 - 7*, 2000, pp. 219–225.

[27] E. Eleftheriou and W. Hirt, "Noise-predictive maximum-likelihood (NPML) detection for the magnetic recording channel," in *Proc. of the IEEE Intern. Conf. on Communications, June 23 – 27, Dallas, TX, USA*, vol. 1, June 1996, pp. 556–560.

[28] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1518–1569, July 2002, special issue on "Shannon Theory: Perspective, Trends, and Applications", in memory of Aaron D. Wyner.

[29] P. D. Feigin and R. L. Tweedie, "Random coefficient autoregressive processes: a Markov chain analysis of stationarity and finiteness of moments," *J. Time Series Analysis*, vol. 6, no. 1, pp. 1–14, 1985.

[30] W. Feller, *An Introduction to Probability Theory and Its Applications.* New York: John Wiley & Sons Inc., 1966, vol. II.

[31] T. R. M. Fischer, "Some remarks on the role of inaccuracy in Shannon's theory of information transmission," in *Proc. 8th Conf. Inform. Theory*, 1971, pp. 22–226.

[32] G. D. Forney, "Codes on graphs: normal realizations," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 520–548, 2001.

[33] C. A. French and J. K. Wolf, "Bounds on capacity of a peak power constrained Gaussian channel," *IEEE Trans. Magnetics*, pp. 2247–2262, Sept. 1998.

[34] R. G. Gallager, *Information Theory and Reliable Communication.* New York: Wiley, 1968.

[35] A. Ganti, A. Lapidoth, and İ. E. Telatar, "Mismatched decoding revisited: General alphabets, channels with memory, and the wide-band limit," *IEEE Trans. Inform. Theory*, pp. 2315–2328, Nov. 2000.

[36] J. S. Goldberg and J. K. Wolf, "Implementation and analysis of nonlinear effects in the micro-track model," *IEEE Trans. Magnetics*, vol. 35, no. 5, pp. 2256–2258, Sept. 1999.

[37] A. J. Goldsmith and P. P. Varaiya, "Capacity, mutual information, and coding for finite-state Markov channels," *IEEE Trans. Inform. Theory*, vol. 42, no. 3, pp. 868–886, May 1993.

[38] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*, 3rd ed. Oxford University Press, Oxford, 2001.

[39] C. Heegard and A. Duel-Hallen, "On the capacity of the noisy run-length channel," in *Proc. of the 1998 IEEE International Workshop in Information Theory, Bejing, China*, July 1988.

[40] C. Heegard, A. Duel-Hallen, and R. Krishnamoorthy, "On the capacity of the noisy runlength channel," *IEEE Trans. Inform. Theory*, vol. 37, no. 3, pp. 712–720, May 1991.

[41] W. Hirt, "Capacity and information rates of discrete-time channels with memory," ETH-Diss. no. 8671, Swiss Federal Institute of Technology (ETH), Zurich, Oct. 1988.

[42] J. L. Holsinger, "Digital communication over fixed time-continuous channels with memory — with special application to telephone channels," *Technical Report No. 430, Laboratory of Electronics, MIT; Lincoln Laboratory, Technical Report 366, Cambridge, MA, USA*, Oct. 20 1964.

[43] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University press, 1999.

[44] D. Hösli and E. Svensson, "Low-density parity-check codes for magnetic recording," Diploma thesis no. 7103, Signal and Information Processing Lab., ETH Zurich, Switzerland, Mar. 2000.

[45] K. A. S. Immink, *Coding techniques for digital recorders*. Prentice Hall, 1991.

[46] K. A. S. Immink, P. H. Siegel, and J. K. Wolf, "Codes for digital recorders," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2260–2299, Oct. 1998.

[47] A. Kavčić, "On the capacity of Markov sources over noisy channels," in *Proc. of the IEEE GLOBECOM 2001, Nov. 25 – 29, San Antonio, USA*, vol. 5, 2001, pp. 2997–3001.

[48] A. Kavčić and A. Parapoutian, "A signal-dependent autoregressive channel model," in *IEEE INTERMAG Conference, Kyongju, Korea*, May 1999.

[49] A. Kavčić and M. Srinivasan, "The minimum description length principle for modeling recording channels," *IEEE J. Sel. Areas Comm.*, vol. 19, pp. 719–729, Apr. 2001.

[50] H. Kobayahsi and D. T. Tang, "Application of partial-response channel coding to magnetic recording systems," *IBM J. Res. Dev.*, pp. 368–375, July 1970.

[51] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[52] B. G. Leroux, "Maximum-likelihood estimation for hidden Markov models," *Stochastic Processes and their Applications*, vol. 40, pp. 127–143, 1992.

[53] H.-A. Loeliger, "Least squares and Kalman filtering on Forney graphs," *in* Codes, Graphs, and Systems, *(Festschrift in honor of David Forney on the occasion of his* 60^{th} *birthday)*, pp. 113–135, 2002, R. E. Blahut and R. Koetter, eds., Kluwer.

[54] X. Ma and A. Kavčić, "Path partitions and forward-only trellis algorithms," *IEEE Trans. Inform. Theory*, vol. 49, no. 1, pp. 38–52, Jan. 2003.

[55] E. Masry, "On covariance functions of unit processes," SIAM J. Appl. Math., vol. 23, no. 1, pp. 28–33, July 1972.

[56] B. McMillan, "History of a problem," in *J. Soc. Indust. Appl. Math.*, vol. 3, 1955, pp. 114–128.

[57] M. Mushkin and I. Bar-David, "Capacity and coding for the Gilbert-Elliot channel," *IEEE Trans. Inform. Theory*, pp. 1277–1290, Nov. 1989.

[58] T. R. Oenning and J. Moon, "The effect of jitter noise on binary input intersymbol interference channel capacity," in *Proc. of the IEEE Intern. Conf. on Communications, June 11 – 14, Helsinki, Finland*, vol. 8, 2001, pp. 2416–2420.

[59] H. D. Pfister, "On the capacity of finite state channels and the analysis of convolutional accumulate-m codes," PhD thesis, University of California, San Diego, 2003.

[60] H. D. Pfister, J. B. Soriaga, and P. H. Siegel, "On the achievable information rates of finite-state ISI channels," in *Proc. of the IEEE GLOBECOM 2001, Nov. 25 – 29, San Antonio, USA*, vol. 5, 2001, pp. 2992–2996.

[61] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.

[62] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[63] A. Ralston and P. Rabinowitz, *A First Course in Numerical Analysis*. International student's edition, McGraw-Hill, 1978.

[64] Sh. Shamai and I. Bar-David, "Information rates for magnetic recording channels with peak- and slope-limited magnetization," *IEEE Trans. Inform. Theory*, vol. 35, no. 5, pp. 956–962, Sept. 1989.

[65] ——, "Upper bounds on capacity for a constrained Gaussian channel," *IEEE Trans. Inform. Theory*, vol. 35, no. 5, pp. 1079–1084, Sept. 1989.

[66] Sh. Shamai and Y. Kofman, "On the capacity of binary and Gaussian channels with run-length-limited inputs," *IEEE Trans. Communications*, vol. 38, no. 5, pp. 584–594, May 1990.

[67] Sh. Shamai and R. Laroia, "The intersymbol interference channel: Lower bounds on capacity and channel precoding loss," *IEEE Trans. Inform. Theory*, vol. 42, no. 5, pp. 1388–1404, Sept. 1996.

[68] Sh. Shamai, L. H. Ozarow, and A. D. Wyner, "Information rate for a discrete-time Gaussian channel with intersymbol interference and stationary inputs," *IEEE Trans. Inform. Theory*, vol. 37, no. 6, pp. 1527–1539, Nov. 1991.

[69] Sh. Shamai and E. Zehavi, "Bounds on the capacity of the bit-shift magnetic recording channel," *IEEE Trans. Inform. Theory*, vol. 37, pp. 863–872, May 1991.

[70] C. Shannon, "A mathematical theory of communication," Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, July and October 1948, reprinted in *Claude Elwood Shannon: Collected Papers*, pp. 5–83, (N. J. A. Sloane and A. D. Wyner, eds.) Piscataway: IEEE Press, 1993.

[71] V. Sharma and S. K. Singh, "Entropy and channel capacity in the regenerative setup with applications to Markov channels," in *Proc. of the IEEE Intern. Symp. on Inform. Theory, June 24 – 29, Washington, USA*, 2001, p. 283.

[72] L. A. Shepp, "Covariance of unit processes," in *Proc. Working Conf. Stochastic Processes, Santa Barbara, CA*, 1967, pp. 205–218.

[73] F. Topsøe, "An information-theoretical identity and a problem involving capacity," *Studia Scientiarum Mathematicarum Hungarica 2*, pp. 291–292, 1967.

[74] P. O. Vontobel, "A generalized Blahut-Arimoto algorithm," Signal and Information Processing Lab., Swiss Federal Institute of Technology (ETH), Zurich, Tech. Rep. INT/200203, Nov. 2002.

[75] P. O. Vontobel and D. M. Arnold, "An upper bound on the capacity of channels with memory and constraint input," in *Proc. of the IEEE Information Theory Workshop, Sept. 2 –7, Cairns, Australia*, 2001, pp. 147–149.

[76] M. L. Williams and R. L. Comstock, "An analytical model for the write process in digital magnetic recording," in *AIP Conf.*, vol. 5, 1971, p. 738.

[77] J. Wolfowitz, "Memory increases capacity," *Information and Control*, vol. 11, no. 4, pp. 423–428, Oct. 1967.

[78] Z.-N. Wu, S. Lin, and J. M. Cioffi, "Numerical results on capacity bounds for magnetic recording," in *Proc. of the IEEE GLOBECOM 1998, Nov. 8 – 12, Sydney, Australia*, 1998, pp. 3385–3390.

[79] E. Zehavi and J. K. Wolf, "On runlength codes," *IEEE Trans. Inform. Theory*, vol. 34, no. 1, pp. 45–54, Jan. 1988.

Index

If an index entry has references to mutiple pages, numbers written in italic refer to the page where the corresponding entry is described.

T

U

About the Author

Dieter M. Arnold was born in Aarau in 1971. After visiting primary school in Luzern, he attended the Gymnasium Alpenquai Luzern, Switzerland, where he received a Matura Typus B (with Latin). In 1991 he joined the Swiss Federal Institute of Technology Zurich (ETH) to study electrical engineering. He graduated with a Diploma degree in electrical engineering from ETH in 1996 after having performed his diploma project at the University of Texas at San Anontio (USA). After one year in industry with ABB in Baden, Switzerland, where he was doing a job-rotation program in the area of high-speed busbar protection, he returned to ETH, where he started as a teaching assistant at the Signal and Information Processing Laboratory (ISI) in 1997. From 1999 to 2002 he was a pre-doctoral student at the IBM Zurich Research Laboratory working in the area of magnetic recording. He received his post-diploma degree in information technology and his PhD both from ETH in 2002 and 2003 respectively.