On Rate-Compatible Insertion Convolutional Turbo Codes and HARQ for Mobile Communications

Von der Fakultät für Elektrotechnik und Informationstechnik
der Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines Doktors der
Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Ingenieur

Tobias Breddermann

aus Aachen

Berichter: Universitätsprofessor Dr.-Ing. Peter Vary
Universitätsprofessor Dr.-Ing. Gerd Ascheid

Tag der mündlichen Prüfung: 02.09.2013

AACHENER BEITRÄGE ZU DIGITALEN NACHRICHTENSYSTEMEN

Herausgeber:

Prof. Dr.-Ing. Peter Vary
Institut für Nachrichtengeräte und Datenverarbeitung
Rheinisch-Westfälische Technische Hochschule Aachen
Muffeter Weg 3a
52074 Aachen
Tel.: 0241-80 26 956
Fax.: 0241-80 22 186

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar

1. Auflage Aachen:
Wissenschaftsverlag Mainz in Aachen
(Aachener Beiträge zu digitalen Nachrichtensystemen, Band 35)
ISSN 1437-6768
ISBN 978-3-86130-288-9

Wissenschaftsverlag Mainz
Süsterfeldstr. 83, 52072 Aachen
Tel.: 02 41 / 2 39 48 oder 02 41 / 87 34 34
Fax: 02 41 / 87 55 77
www.Verlag-Mainz.de

Herstellung: Druckerei Mainz GmbH,
Süsterfeldstr. 83, 52072 Aachen
Tel.: 02 41 / 87 34 34; Fax: 02 41 / 87 55 77
www.druckereimainz.de

Gedruckt auf chlorfrei gebleichtem Papier

D 82 (Diss. RWTH Aachen University, 2013)

Acknowledgments

This thesis was written during my time as research assistant at the *Institute of Communication Systems and Data Processing* (IND) at the *Rheinisch-Westfälische Technische Hochschule Aachen* (RWTH Aachen University).

My special thanks goes to my supervisor Prof. Dr.-Ing. Peter Vary for his continuous support of my research by fruitful discussions we had over the years. His ideas and suggestions have strongly contributed to the success of my work. I am also indebted to my co-supervisor Prof. Dr.-Ing. Gerd Ascheid for showing his interest in my work.

Furthermore, I want to express my gratitude to all of my current and former colleagues as well as the permanent staff at the IND for providing a very pleasant and enjoyable working environment. In particular, I am grateful to Dr.-Ing. Christiane Antweiler, Dipl.-Ing. Moritz Beermann, Dipl.-Ing. Benedikt Eschbach, Dr.-Ing. Bernd Geiser, Dipl.-Ing. Florian Heese, Dipl.-Ing. Matthias Pawig, Dipl.-Ing. Matthias Rüngeler, Dipl.-Ing. Magnus Schäfer, Dipl.-Ing. Thomas Schlien, and Dipl.-Inform Florian Schmidt for their proof-readings which have resulted in valuable improvements of this thesis.

I would also like to thank the *Ulta high-speed Mobile Information and Communication* (UMIC) research cluster of excellence for financing the project I was involved in.

Finally, I owe my loving thanks to my parents Maria and Rudolf Breddermann as well as my sister Annika for their personal support which made it possible to arrive at this stage. Last but not least, I am deeply grateful to my wonderful wife Kirsten. Thank you for your love, encouragement, patience, and the caring education of our lovely children Sophie and Elisa.

Lippstadt, October 2013 Tobias Breddermann

Abstract

Rate-compatible Turbo codes are part of many mobile communication systems as, e.g., LTE. The code rate of a fixed-rate mother Turbo code is adapted by bit puncturing and bit repetition. While bit puncturing allows to flexibly increase the effective code rate by discarding a certain number of encoded bits, bit repetition facilitates lower effective code rates by duplicating a distinct number of encoded bits prior to transmission.

The first part of this thesis deals with dummy bit insertion, which is a high potential alternative to bit repetition. Lower code rates can be constructed by inserting known (dummy) bits into the information bit sequence before Turbo coding (referred to as *Rate-Compatible Insertion Convolutional* (RCIC) Turbo coding). Besides an improved frame error rate performance, a considerable convergence speedup is realized for LTE. These observations are also confirmed by means of their EXIT charts, which have been derived from the EXIT chart of the mother Turbo code using information theoretic models. Similar derivations are also given for Turbo coding with bit puncturing and bit repetition. In all cases, the obtained expressions result in novel semi-analytical rate matching procedures, which determine analytically the optimal code rate required for (nearly) error-free decoding at a specific target channel quality. A successful application of dummy bit insertion to *Low-Density Parity-Check* (LDPC) codes is also provided in this part.

The second part of the thesis focuses on *Hybrid Automatic Repeat reQuest* (HARQ). At first, a novel HARQ scheme for LTE based on bit puncturing and dummy bit insertion is proposed. Evaluations of the system throughput have proven that the novel system surpasses the standardized LTE HARQ system in a wide range of channel conditions. Then, HARQ with unreliable feedback is studied. Based on a general feedback channel model, analytical expressions for the overall system throughput are derived in dependence of the feedback error probabilities and the frame error rates of the HARQ system with reliable feedback. Extensive time-consuming simulations of the HARQ system with unreliable feedback can, thus, be avoided.

Finally, *Header bit Assisted Channel Decoding* (HACD) is addressed. Upper bounds on the expected gains are determined analytically based on the EXIT chart analysis presented in the first part of this thesis.

Contents

List of Abbreviations

3GPP	*3rd Generation Partnership Project*
ACK	*ACKnowledge*
APP	*Application Layer*
AWGNC	*Additive White Gaussian Noise Channel*
BCJR	*Bahl-Cocke-Jelinek-Raviv*
BEC	*Binary Erasure Channel*
BER	*Bit Error Rate*
BIAWGNC	*Binary-Input Additive White Gaussian Noise Channel*
BICM-ID	*Bit Interleaved Coded Modulation with Iterative Decoding*
BPSK	*Binary Phase Shift Keying*
BSC	*Binary Symmetric Channel*
CRC	*Cyclic Redundancy Check*
DBI	*Dummy Bit Insertion*
DVB-S	*Digital Video Broadcast - Satellite*
DVB-T	*Digital Video Broadcast - Terrestrial*
EEP	*Equal Error Protection*
EXIT	*EXtrinsic Information Transfer*
FEC	*Forward Error Correction*
FER	*Frame Error Rate*
HACD	*Header bit Assisted Channel Decoding*
HARQ	*Hybrid Automatic Repeat reQuest*
HSDPA	*High Speed Downlink Packet Access*
HSPA	*High Speed Packet Access*
HSUPA	*High Speed Uplink Packet Access*
IHCD	*Iterative Header-Channel Decoding*
IMT-Advanced	*International Mobile Telecommunications - Advanced*

IP	*Internet Protocol*
IR	*Incremental Redundancy*
ISCD	*Iterative Source-Channel Decoding*
ISO	*International Organization for Standardization*
ITU	*International Telecommunication Union*
ITU-T	*ITU Telecommunication Standardization Sector*
JPCD	*Joint Protocol-Channel Decoding*
LDPC	*Low-Density Parity-Check*
LLR	*Log-Likelihood Ratio*
LogMAP	symbol-by-symbol *Maximum A Posteriori* estimation in the *Logarithmic domain*
LTE	*Long Term Evolution*
MAC	*Medium Access Control*
MAP	*Maximum A Posteriori*
MCS	*Modulation and Coding Scheme*
MMSE	*Minimum Mean Square Error*
NACK	*Negative ACKnowledge*
OFDM	*Orthogonal Frequency-Devision Multiplexing*
OSI	*Open Systems Interconnection*
PCCC	*Parallel Concatenated Convolutional Code*
PDCP	*Packet Data Convergence Protocol*
PHY	*Physical Layer*
QAM	*Quadrature Amplitude Modulation*
QoS	*Quality of Service*
QPSK	*Quadrature Phase Shift Keying*
RCC	*Rate-Compatible Convolutional*
RCIC	*Rate-Compatible Insertion Convolutional*
RCPC	*Rate-Compatible Punctured Convolutional*
RCRC	*Rate-Compatible Repetition Convolutional*
RLC	*Radio Link Control*
RNSC	*Recursive Non-Systematic Code*
RoHC	*Robust Header Compression*
RTP	*Real-time Transport Protocol*
RV	*Redundancy Version*

SCCC	*Serially Concatenated Convolutional Code*
SDHD	*Soft Decision Header Decoding*
SDSD	*Soft Decision Source Decoding*
SISO	*Soft-In/Soft-Out*
SNR	*Signal-to-Noise Ratio*
TTI	*Transmission Time Interval*
UDP	*User Datagram Protocol*
UEP	*Unequal Error Protection*
UMTS	*Universal Mobile Telecommunications System*
WiMAX	*Worldwide Interoperability for Microwave Access*
WLAN	*Wireless Local Area Network*

List of Symbols

$\mathbf{A}$	parity-check matrix of an LDPC code
α	initial state reliability for symbol-by-symbol MAP estimation
β	final state reliability for symbol-by-symbol MAP estimation
$\mathbf{b}$	information bit vector, input of channel encoder
$\hat{\mathbf{b}}$	estimated information bit vector
$\mathbf{c}$	parity bit vector generated by the channel encoder
$\mathfrak{c}$	check-to-variable node message
$\mathbf{d}$	vector of dummy bits
$\mathbf{e}$	output bit vector of the outer channel encoder (SCCCs)
E_b	energy per information bit
E_s	energy per modulation symbol
$\mathcal{G}$	maximum throughput (envelope) of all specified MCSs
γ	state transition reliability for symbol-by-symbol MAP est.
$\mathbf{G}_{\mathrm{CC}}$	generator polynomial(s) of a convolutional code
$\mathbf{G}_{\mathrm{in}}$	generator polynomial(s) of an inner convolutional code
$\mathcal{G}_{\mathrm{mcs}}$	throughput of a specific MCS
$\mathbf{G}_{\mathrm{out}}$	generator polynomial(s) of an outer convolutional code
$\mathbf{h}$	vector of header bits
H'	binary entropy function
I	number of bits per modulation symbol
$\mathcal{I}$	mutual information
$\mathcal{I}_{\mathrm{C}}$	mutual information of the communication channel
$\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}$	mutual *a priori* information at channel dec. input (PCCCs)
$\mathcal{I}_{\mathrm{CD}}^{[\mathrm{ext}]}$	mutual *extrinsic* information at channel dec. output (PCCCs)
$\mathcal{I}_{\mathrm{CN}}^{[\mathrm{apri}]}$	mutual *a priori* information at check node decoder input
$\mathcal{I}_{\mathrm{in}}^{[\mathrm{apri}]}$	mutual *a priori* information at inner dec. input (SCCCs)

$\mathcal{I}_{\text{in}}^{[\text{ext}]}$	mutual *extrinsic* information at inner dec. output (SCCCs)
$\mathcal{I}_{\text{out}}^{[\text{apri}]}$	mutual *a priori* information at outer dec. input (SCCCs)
$\mathcal{I}_{\text{out}}^{[\text{ext}]}$	mutual *extrinsic* information at outer dec. output (SCCCs)
J	ten Brink's J-function
$\mathfrak{L}$	constraint length of a convolutional code
K	maximum number of HARQ transmissions
$\overline{K}$	average number of HARQ transmissions
λ	variable node degree distribution (node perspective)
λ'	variable node degree distribution (edge perspective)
ℓ_b	number of information bits
ℓ_c	number of parity bits generated by the channel encoder
$\mathbf{L}_{\text{CD}}$	vector of *a posteriori* LLRs at channel dec. output (PCCCs)
$\mathbf{L}_{\text{CD}}^{[\text{apri}]}$	vector of *a priori* LLRs at channel decoder input (PCCCs)
$\mathbf{L}_{\text{CD}}^{[\text{ext}]}$	vector of *extrinsic* LLRs at channel decoder output (PCCCs)
$\mathbf{L}_{\text{CN}}^{[\text{ext}]}$	vector of *extrinsic* LLRs determined at check nodes
ℓ_d	number of inserted dummy bits
$\mathbf{L}_{\text{DM}}^{[\text{chan}]}$	vector of channel-related LLRs at soft demodulator output
ℓ_e	number of code bits generated by outer encoder (SCCCs)
ℓ_h	number of header bits
$\mathbf{L}_{\text{HD}}^{[\text{ext}]}$	vector of *extrinsic* LLRs at header decoder output
$\mathbf{L}_{\text{in}}^{[\text{apri}]}$	vector of *a priori* LLRs at inner decoder input (SCCCs)
$\mathbf{L}_{\text{in}}^{[\text{ext}]}$	vector of *extrinsic* LLRs at inner decoder output (SCCCs)
$\mathbf{L}_{\text{out}}$	vector of *a posteriori* LLRs at outer dec. output (SCCCs)
$\mathbf{L}_{\text{out}}^{[\text{apri}]}$	vector of *a priori* LLRs at outer decoder input (SCCCs)
$\mathbf{L}_{\text{out}}^{[\text{ext}]}$	vector of *extrinsic* LLRs at outer decoder output (SCCCs)
ℓ_s	number of time samples per multimedia frame
$\mathbf{L}_{\text{SD}}^{[\text{ext}]}$	vector of *extrinsic* LLRs at source decoder output
ℓ_v	number of bits after source encoding (payload)
$\mathbf{L}_{\text{VN}}$	vector of *a posteriori* LLRs determined at variable nodes
$\mathbf{L}_{\text{VN}}^{[\text{ext}]}$	vector of *extrinsic* LLRs determined at variable nodes
ℓ_x	number of bits after dummy bit insertion
ℓ_y	number of code bits
N_0	spectral noise power density
P_0	cross-over probability of a BSC
P_{A}	probability for ACK→NACK error
P_{C}	erasure probability of communication channel (BEC)

P_{D}	inserted dummy bit fraction
P_{E}	erasure probability of a BEC
P_i	FER after i-th HARQ transmission
$P_{i\|i}$	probability of i HARQ transmission if i transmissions allowed
P_{N}	probability for NACK→ACK error
r_{CC}	code rate of a convolutional code
r_{DR}	repetition rate of a deterministic repeater
ϱ	check node degree distribution (node perspective)
ϱ'	check node degree distribution (edge perspective)
r_{IC}	code rate of an insertion convolutional code
r_{in}	code rate of a convolutional code employed as inner code
r_{LDPC}	code rate of an LDPC code
r_{out}	code rate of a convolutional code employed as outer code
r_{P}	puncturing rate
r_{PC}	code rate of a punctured convolutional code
r_{R}	repetition rate
r_{RC}	code rate of a repetition convolutional code
r_{RM}	effective code rate after rate matching
r_{RP}	puncturing rate of a random puncturer
$\mathbf{S}$	vector of complex modulation symbols
$\mathbf{s}$	vector of time samples before source encoding
$\hat{\mathbf{s}}$	vector of estimated time samples after source decoding
$\sigma_{\mathcal{L}}^2$	variance of LLRs
σ^2	variance of received signal
$\mathcal{T}$	EXIT function
$\mathcal{T}_{\mathrm{CD}}$	EXIT function of a component code in PCCC schemes
$\mathcal{T}_{\mathrm{CD}}^{(\mathrm{IC})}$	EXIT function of an RCIC component code (PCCC)
$\mathcal{T}_{\mathrm{CD}}^{(\mathrm{PC})}$	EXIT function of an RCPC component code (PCCC)
$\mathcal{T}_{\mathrm{CD}}^{(\mathrm{RC})}$	EXIT function of an RCRC component code (PCCC)
$\mathcal{T}_{\mathrm{CN}}$	EXIT function of a check node decoder
$\mathcal{T}_{\mathrm{in}}$	EXIT function of an inner code in SCCC schemes
$\mathcal{T}_{\mathrm{in}}^{(\mathrm{IC})}$	EXIT function of an RCIC inner code (SCCC)
$\mathcal{T}_{\mathrm{in}}^{(\mathrm{PC})}$	EXIT function of an RCPC inner code (SCCC)
$\mathcal{T}_{\mathrm{in}}^{(\mathrm{RC})}$	EXIT function of an RCRC inner code (SCCC)
$\mathcal{T}_{\mathrm{out}}$	EXIT function of an outer code in SCCC schemes
$\mathcal{T}_{\mathrm{out}}^{(\mathrm{PC})}$	EXIT function of an RCPC outer code (SCCC)
$\mathcal{T}_{\mathrm{out}}^{(\mathrm{RC})}$	EXIT function of an RCRC outer code (SCCC)
$\mathfrak{v}$	variable-to-check node message

$\mathbf{v}$	vector of payload bits
$\mathbf{x}$	input vector of the channel encoder after dummy bit insertion
$\mathbf{y}$	bit vector after channel encoding
$\check{\mathbf{y}}$	bit vector after rate matching
$\dot{\mathbf{y}}$	vector of punctured code bits
$\mathbf{Z}$	vector of demodulated bits distorted by channel noise

Chapter 1

Introduction

This thesis has been created within the *Ultra high-speed Mobile Information and Communication* (UMIC) cluster of excellence, which is an interdisciplinary academic research cluster established under the German Federal and State Government Excellence Initiative. Driven by the rapidly growing user demands on mobile communication systems, UMIC aims at developing new fundamental concepts and paradigms for future mobile access with a strong focus on providing significantly higher data rates.

The possibility of having any information available nearly everywhere and at any time has led to a dramatic increase of the mobile data traffic. Cisco, one of the worldwide leader for network solutions, e.g., assumes in its *Global Mobile Data Traffic Forecast Update, 2011–2016* [Cis12] an increase of the world's data traffic from $0.6 \cdot 10^9$ GB (Gigabyte) per month in 2011 to $10.8 \cdot 10^9$ GB per month in 2016, which is a relative increase of 1800 %. As predicted in this and other forecasts, there is still a strong demand for higher data rates which affects the organization of future networks and the realization of their basic components. It is ambitious to strive for much higher data rates under the constraint of a moderately growing frequency spectrum used for transmission. Thus, a more efficient use of this limited resource within the whole network is indispensable.

This thesis concentrates on physical layer aspects. The physical layer in mobile communication systems is mainly responsible for the transmission with protection against bit errors caused by various impairments on the wireless link and has a deep impact on the spectral efficiency of the system. Common error protection schemes (channel coding schemes) used in state-of-the-art systems are optimized for specific target channel qualities resulting in a considerable loss in spectral efficiency at other channel qualities. However, mobile communication systems have to operate at a wide range of channel conditions, since users at the boundary of a radio cell and users near the base station experience entirely different conditions of the radio channel. To allow for a sufficient coverage in the radio cells, error protection is, therefore, carried out by means of *adaptive* modulation and *adaptive* coding (rate matching) attempting to approach the theoretical limits [Sha48] for the complete range of target channel qualities. While adaptive modulation switches the employed modulation scheme, rate matching allows the system to adapt the

code rate during transmission with respect to the instantaneous channel condition. The analysis and development of a novel rate matching technique with increased effectiveness as well as its comparison to standardized techniques is the major focus of this thesis.

Rate matching schemes specified in standardized communication systems pursue different strategies. In the IEEE 802.16e (WIMAX) standard [IEE04] and the IEEE 802.11n (WLAN) standard [IEE09], rate matching is realized by ensembles of LDPC codes [Gal62, MWD99] with individual (different) code rates. The component code which is most likely to provide the highest system throughput under the given channel condition is selected for transmission. In other systems, e.g., UMTS LTE[1] [3GP11], channel coding is performed by a fixed-rate, single mother Turbo code [BGT93, BG96] followed by a low-complexity rate matching scheme which adapts the code rate of the mother code to the desired target code rate. This provides a higher code rate flexibility, a decreased computational complexity, and enables highly optimized encoder and decoder implementations since only one encoder/decoder pair needs to be employed. Channel codes which exhibit such a rate-flexibility are commonly called *rate-compatible* codes.

Rate-compatible codes have initially been designed for convolutional codes and are now widely used in systems based on Turbo codes as, e.g., in the LTE system. Bit repetition and bit puncturing (bit canceling) are commonly employed after fixed-rate convolutional (Turbo) coding to provide lower and higher code rates, respectively. Usually, these codes are referred to as *Rate-Compatible Repetition Convolutional* (RCRC) codes [LS00] and *Rate-Compatible Punctured Convolutional* (RCPC) codes [CCG79, Hag88]. It has further been discovered that rate-compatible codes are well suitable for HARQ type-II schemes [Man74] in which the transmission of *additional* redundancy about the same data frame can be requested by the receiver. The transmitter stops the transmissions of new frames until either decoding is possible or the maximum accepted number of transmissions per data frame is reached. Such a *Redundancy Version* (RV) can contain, e.g., bits which have initially been punctured, and/or duplicates of already transmitted bits. All RVs are finally combined at the receiver before channel decoding leading to a significantly reduced number of frame losses. Those schemes are part of the most standardized wireless and mobile communication systems.

An alternative class to RCRC codes has been presented in [XR00]. This class can simply construct lower rate codes from a fixed rate, single mother code by inserting known bits (dummy bits) into the information bit sequence *before* convolutional encoding. Throughout this thesis, these codes are referred to as RCIC codes. Other common names are *determinate state* convolutional codes [CH93] or *pruned* convolutional codes [WC98]. Although, RCIC (Turbo) codes are not

[1] In the following, the term LTE is used instead of the original term UMTS LTE. However, note that LTE only specifies a *long term evolution* of UMTS and does not denote the next (fourth) mobile generation as it is often erroneously assumed.

part of any standards in communication so far, they are high potential candidates for rate matching schemes in future mobile systems and of major interest in this thesis. Furthermore, the underlying concept has been employed to LDPC codes as well.

Structure of the Thesis

In particular in the context of RCIC codes, there are still a number of open issues which are addressed in this work. To this end, the essential concepts and, furthermore, the basic definitions/notations are introduced in the first two chapters.

Chapter 2 describes fundamentals on Turbo codes and LDPC codes, and provides a brief overview about *EXtrinsic Information Transfer* (EXIT) charts [tB01a] which are well suited to predict their convergence behavior. *Chapter 3* continues with an introduction of different classes of rate-compatible codes which have been considered in this work. This covers the well-known RCPC and RCRC codes, their practical realization in LTE, as well as RCIC codes which are not part of any standards in communication so far.

The main part of the thesis is treated in *Chapter 4*. Although the potential of RCIC codes have been demonstrated in several practical scenarios, an information theoretic analysis of the expected gains dependent on the fraction of inserted dummy bits has been missing so far. Therefore, this chapter introduces an appropriate information theoretic transmission model which is capable of analyzing the influence of dummy bit insertion on the convergence behavior of Turbo codes using EXIT charts. This analysis enables the semi-analytical optimization of rate matching by means of dummy bit insertion with respect to its spectral efficiency, avoiding extensive simulations. Only the EXIT chart of the mother code has to be determined by means of simulations. Besides rate matching, it further supports or even simplifies the design of other transmission concepts in which perfect information about a fraction of bits can be exploited. Prominent examples are *joint channel coding and cryptography* [RZ08], *pilot symbol assisted coding* [OKPL08, BCH09b], and *header bit assisted channel decoding* [BLV+10] (addressed in Chapter 7 in more detail). Moreover, a transmission model which covers RCRC codes is also derived in this chapter. This facilitates the comparison of the competing code classes RCIC and RCRC codes. An information theoretic evaluation of RCPC codes has already been presented in [Tho07] and is revisited for the sake of completeness. Since many modern communication systems employ LDPC codes, the information theoretic models are, furthermore, adapted to the LDPC case, increasing the benefit of the presented analysis.

Successful applications of dummy bit insertion to LTE and WiMAX are then demonstrated in *Chapter 5*. For Turbo coding systems (LTE), it is known that dummy bits should be equidistantly distributed within the frame due to the strong dependencies between adjacent bits. However, WiMAX employs LDPC codes

which introduce dependencies according to the entries in the parity-check matrix of the code, each resulting in one connection within the Tanner graph [Tan81]. Therefore, the development of an optimal dummy bit distribution is not as trivial as for convolutional (Turbo) codes. However, there have been proposed several insertion schemes so far which differ in their optimization constraint, e.g, [TJ05, Oka08, LWZ09]. In this thesis, a novel algorithm is proposed in which those positions are selected for dummy bits which have approximately equal distance within the Tanner graph. This follows the argumentation known from RCIC codes.

A further crucial challenge covered by this thesis is the optimal integration of RCIC codes in systems employing HARQ type-II schemes. In contrast to bit repetition and bit puncturing, dummy bit insertion is carried out *before* channel encoding. The fraction of inserted dummy bits is, thus, determined by the code rate of the initial transmission. If the receiver requests additional transmissions due to residual bit errors after decoding, incremental redundancy cannot be generated by dummy bit insertion anymore and the system has to use less effective schemes as, e.g, bit repetition. In the first part of *Chapter 6*, a novel HARQ type-II scheme for LTE based on bit puncturing and dummy bit insertion is proposed which overcomes this drawback. The second part addresses HARQ with *unreliable* feedback. Usually, HARQ schemes are evaluated in terms of system throughput assuming perfect feedback [Kal90, LLMC12], i.e., neglecting the behavior of the wireless feedback link. There are only a few contributions incorporating the impact of unreliable feedback on the system performance, e.g., [Wu10, MCM11]. Moreover, they are mostly restricted to specific channel models or HARQ protocols. Therefore, an analysis of the influence of feedback errors on the system throughput based on a general HARQ model without these restrictions is carried out in this chapter.

In *Chapter 7*, HACD as a novel iterative receiver concept is proposed. In HACD, perfect information about error-free decoded headers is fed back to the channel decoder resulting in an improved decoding of the payload. Since the header bits take the role of dummy bits during the decoding process, a performance bound on the expected gains is derived using the theoretic results presented in Chapter 4. This bound has been very beneficial to evaluate the potential of HACD dependent on the header bit fraction and will highly support the design of these systems.

All the above mentioned issues are picked up in *Chapter 8* again for a final discussion of the essential results.

Most of the presented results have been pre-published in the following references: [BLV+10, BV10a, BV10b, BBV11, BEV11, BV11a, BV11b, STBV11, BEV12a, BEV12b]. These references are underlined, throughout this thesis.

Chapter 2

Packet-Switched Mobile Communication

Mobile and wireless communication systems are nowadays organized as packet-switched networks rather than circuit-switched networks in order to enable the transmission of any data (voice, multimedia content, preprocessed data, ...) between several users using the same network. All these networks are realized in accordance to the *Open Systems Interconnection* (OSI) reference model [ISO96] standardized by the *International Organization for Standardization* (ISO). In general, this model defines seven logically separated layers which exchange information in a well-ordered manner. The exchange of information is managed by layer-specific protocols which define rules for the interpretation and the receiver-related processing of the data. This enables a so-called *horizontal* communication between two instances located at the same layer. Furthermore, the ISO-OSI reference model allows a *vertical* communication between different layers in which lower layers exchange information with higher layers. Communication is only permitted between adjacent layers to ensure the logical separation of the layers. The layered structure of modern communication systems is visible in the general transmission model described in Section 2.1.

2.1 General Transmission System Model

The general transmission model considered throughout this thesis is depicted in Figure 2.1. It is logically divided into three parts: The *Application Layer* (APP) which comprises source-related processing components, the *Intermediate Network Layers* which are mainly responsible for the packet routing through the network, and the *Physical Layer* (PHY) which provides link-related features. The aim of this section is the description of all relevant components and the introduction of the utilized notation.

Transmitter

In heterogeneous networks, the input of the transceiver can be either a multimedia signal carrying, e.g., voice, audio, or video content, or even a frame of preprocessed

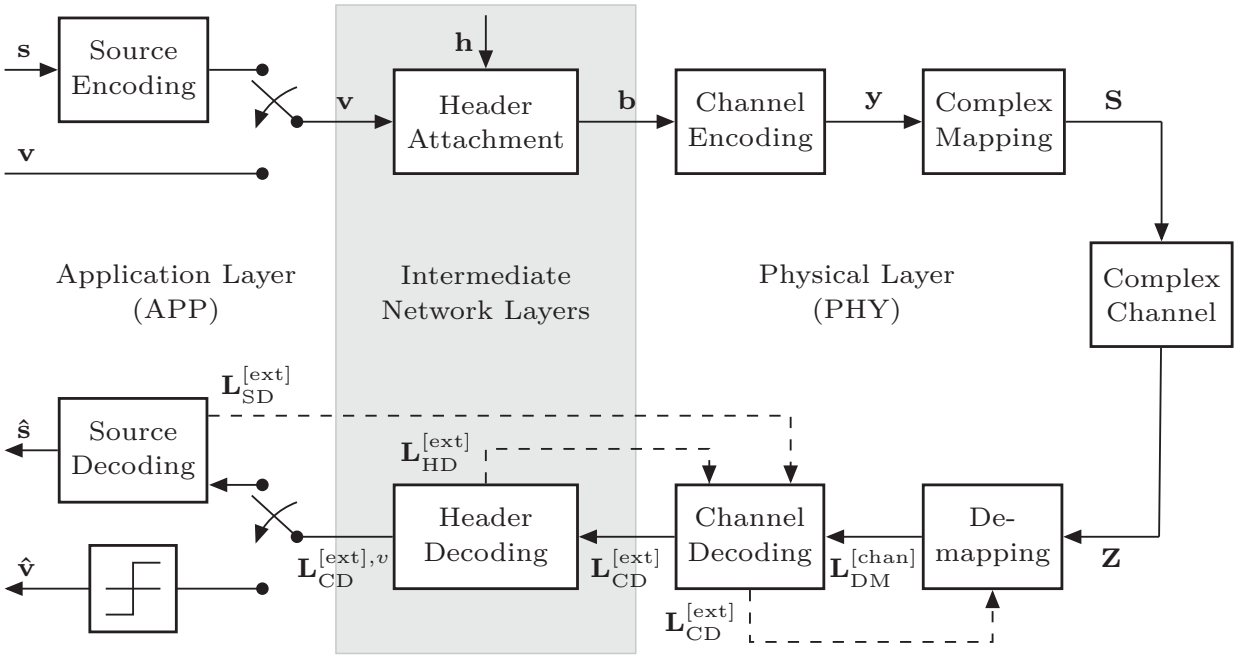


Figure 2.1: General transmission system model for packet-switched mobile communication.

data. In the latter case, the preprocessed bit vector $\mathbf{v} \in \mathbb{F}_2^{\ell_v}$ of length ℓ_v is directly forwarded to the intermediate network layers. In the other cases, source encoding is commonly employed in order to compress the real-valued but discrete time signal $\mathbf{s} \in \mathbb{R}^{\ell_s}$ of length ℓ_s to a bit vector $\mathbf{v} \in \mathbb{F}_2^{\ell_v}$. Then, a header $\mathbf{h} \in \mathbb{F}_2^{\ell_h}$ of length ℓ_h is attached to the payload $\mathbf{v}$ according to $\mathbf{b} = [\mathbf{h}, \mathbf{v}] \in \mathbb{F}_2^{\ell_b}$ with $\ell_b = \ell_h + \ell_v$.

At the PHY, the bit frame $\mathbf{b}$ is protected against transmission errors which are caused by the impairments on the wireless link. A channel encoder with code rate r_{C} adds artificial redundancy to the input frame $\mathbf{b}$ resulting in the encoded output bit frame $\mathbf{y} \in \mathbb{F}_2^{\ell_y}$ with length $\ell_y = r_{\mathrm{C}}^{-1}\ell_b$. In modern wireless and mobile communication systems, Turbo codes [BGT93, BG96] or *Low-Density Parity-Check* (LDPC) codes [Gal62, Mac99] are commonly employed. Prominent examples are UMTS [3GP05], LTE [3GP11], WLAN [IEE09], WiMAX [IEE04], DVB-S2 [ETS09a] or DVB-T2 [ETS09b]. Both code classes are described in Sections 2.2.1 and 2.2.2, respectively, in more detail. It is worth mentioning that rate matching schemes, which are addressed in Chapter 3 for convolutional (Turbo) codes, are also covered by this component.

Prior to transmission, the encoded bits $\mathbf{y}$ are grouped to vectors of I bits which are assigned to complex modulation symbols $\mathbf{S} \in \mathbb{C}^{\ell_y/I}$ from a signal constellation set $\mathbb{S} \subset \mathbb{C}$ according to a specified mapping rule. Note that the baseband modulation including, e.g., spreading or multi-carrier transformation (OFDM), and the radio frequency frontend are left out for the sake of clarity.

Receiver

The receiver receives a distorted and complex-valued symbol vector $\mathbf{Z} \in \mathbb{C}^{\ell_y/I}$ of length ℓ_y/I from the channel. These symbols are then mapped by a soft demapper to a vector $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{y}) \in \mathbb{R}^{\ell_y}$ which contains reliability information on the code bits $\mathbf{y}$ in terms of *Log-Likelihood Ratio*s (LLRs) [HOP96]. Employing a *Soft-In/Soft-Out* (SISO) channel decoder, this information can be exploited to determine so-called *extrinsic* information $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{ext}]}(\mathbf{b}) \in \mathbb{R}^{\ell_b}$ on the information bits $\mathbf{b}$ by means of symbol-by-symbol *Maximum A Posteriori* estimation in the *Logarithmic domain* (LogMAP) decoding [BCJR74, RVH95]. The *extrinsic* information on the header bits $\mathbf{h}$ and the payload bits $\mathbf{v}$ are forwarded to the intermediate network layers. The headers are decoded based on the given reliability information and checked for consistency using the included checksums or an additional *Cyclic Redundancy Check* (CRC). Additionally, *a priori* knowledge about the header fields, i.e., their inherently given redundancy, can support the recovery of the packet headers (see Section 7). If a header cannot be retrieved, the complete packet will be discarded since successful routing is impossible. Otherwise, the vector $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{ext}],v} = \mathbf{L}_{\mathrm{CD}}^{[\mathrm{ext}]}(\mathbf{v})$ is forwarded to the APP. The source decoder considered throughout this thesis is capable of processing soft information and exploiting the source distribution as additional *a priori* information. An estimate $\hat{\mathbf{s}} \in \mathbb{R}^{\ell_s}$ of the input signal $\mathbf{s}$ can then be computed by means of *Maximum A Posteriori* (MAP) or even *Minimum Mean Square Error* (MMSE) estimation. This process is called *Soft Decision Source Decoding* (SDSD) [FV01].

Several loops between two or more components can be carried out by modern receivers. *Bit Interleaved Coded Modulation with Iterative Decoding* (BICM-ID) [LR97], e.g., describes the process in which *extrinsic* information $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{ext}]}(\mathbf{y})$ on the code bits $\mathbf{y}$ is fed back by the SISO channel decoder to improve the channel-related LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{y})$ delivered by the soft demapper in an iterative process. If the source decoder is capable of generating *extrinsic* information $\mathbf{L}_{\mathrm{SD}}^{[\mathrm{ext}]}(\mathbf{v})$ on the payload bits $\mathbf{v}$, *Iterative Source-Channel Decoding* (ISCD) can be carried out [Gör00, AVS01]. If source decoder, channel decoder, and demapper are iteratively concatenated, then, the process is denoted as *Turbo DeCodulation* (TDeC) [CBAV05]. In packet-switched networks, an additional feedback $\mathbf{L}_{\mathrm{HD}}^{[\mathrm{ext}]}(\mathbf{h})$ from the header decoder to the channel decoder is possible. This concept is called *Header bit Assisted Channel Decoding* (HACD) and is addressed in Chapter 7 in more detail.

2.2 Channel Coding Concepts

Modern communication systems employ Turbo codes or LDPC codes for physical layer channel coding. The fundamentals of both code classes are briefly described in Sections 2.2.1 and 2.2.2. Furthermore, EXIT charts are introduced in Section 2.2.3 as a powerful analysis tool which is used throughout this thesis.

2.2.1 Turbo Codes

The Turbo principle has initially been developed for *Parallel Concatenated Convolutional Code*s (PCCCs) by Berrou in 1993 [BGT93, BG96]. It was the first verification that channel decoding near the theoretical performance limit [Sha48] is possible with moderate complexity. In the following years, this concept was adapted to serially concatenated coding schemes in which block codes as well as convolutional codes were employed as component codes [BM96a, BM96b]. However, we restrict ourselves to the consideration of PCCCs and *Serially Concatenated Convolutional Code*s (SCCCs) throughout this section.

Parallel Concatenated Convolutional Codes

The encoder of a parallel concatenated convolutional code is depicted in Figure 2.2(a). The information bit vector $\mathbf{b} = (b_1, \ldots, b_m, \ldots, b_{\ell_b})$ is separately encoded by two convolutional component codes, each generating ℓ_c parity bits $\mathbf{c}^I = (c_1^I, \ldots, c_i^I, \ldots, c_{\ell_c}^I)$ and $\mathbf{c}^{II} = (c_1^{II}, \ldots, c_i^{II}, \ldots, c_{\ell_c}^{II})$, respectively. As shown in Figure 2.2(a), the information bits $\mathbf{b}$ are interleaved by an interleaver π before encoding with Encoder II. Pseudo-random interleavers show considerable performance improvements compared to block interleavers [HLY02] and are, therefore,

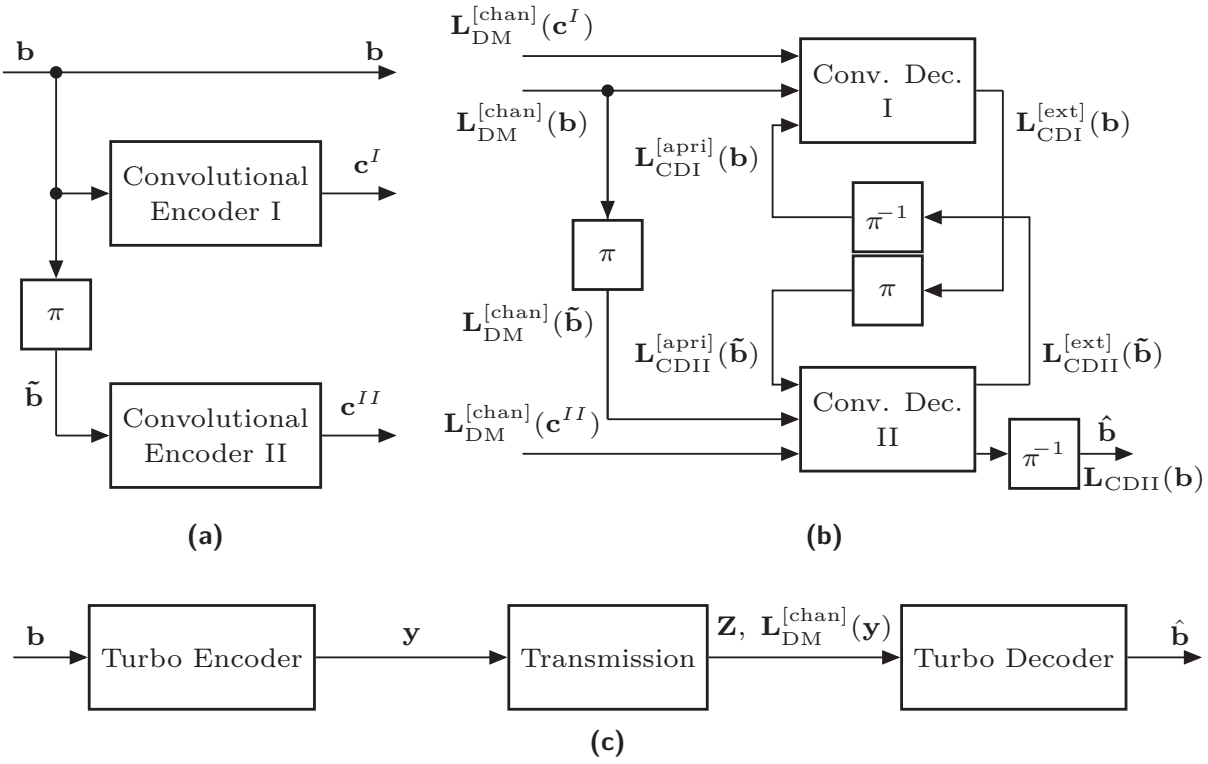


Figure 2.2: Turbo encoder (a) and Turbo decoder (b) for a parallel concatenated convolutional code. A simplified transmission chain is shown in (c).

commonly employed. The output bit streams are finally multiplexed to the overall output vector $\mathbf{y} = (y_1, \ldots, y_n, \ldots, y_{\ell_y})$ as shown in Figure 2.2(c).

After bipolar transmission, i.e., $y_n \in \{-1, 1\}$, channel-related *a posteriori* LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{y}) = (L_{\mathrm{DM}}^{[\mathrm{chan}]}(y_1), \ldots, L_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n), \ldots L_{\mathrm{DM}}^{[\mathrm{chan}]}(y_{\ell_y}))$ are computed based on the complex channel observation vector $\mathbf{Z}$ according to

$$L_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n) := L(y_n|\mathbf{Z}) = \ln\left(\frac{P(y_n = 1|\mathbf{Z})}{P(y_n = -1|\mathbf{Z})}\right) \tag{2.1}$$

and provided to the Turbo decoder. For BPSK transmission over a fading *Additive White Gaussian Noise* (AWGN) channel and equiprobable bits y_n, this expression can be determined in dependence of the fading factor a and the channel *Signal-to-Noise Ratio* (SNR) E_s/N_0 [HOP96]:

$$L_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n) = \ln\left(\frac{P(Z_n|y_n = 1)}{P(Z_n|y_n = -1)}\right) = 4 \cdot a \cdot \frac{E_s}{N_0} \cdot Z_n, \quad Z_n \in \mathbb{R}. \tag{2.2}$$

If the communication channel can be modeled by a *Binary Erasure Channel* (BEC), with $Z_n \in \{-1, *, 1\}$ and $Z_n = *$ denoting a bit erasure, or a *Binary Symmetric Channel* (BSC) with $Z_n \in \{-1, 1\}$ and crossover probability $P_0 \leq 1/2$, these LLRs are given by:

$$\text{BEC:} \quad L_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n) = \begin{cases} \infty \cdot Z_n: & Z_n \in \{-1, 1\} \\ 0 \quad\quad : & Z_n = * \end{cases} \tag{2.3}$$

$$\text{BSC:} \quad L_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n) = \ln\left(\frac{1 - P_0}{P_0}\right) \cdot Z_n, \quad Z_n \in \{-1, 1\}. \tag{2.4}$$

The reader is referred to Appendix A for more details on the considered channel models.

The Turbo decoder illustrated in Figure 2.2(b) consists of two parallel concatenated SISO channel decoders which compute *extrinsic* information

$$\mathbf{L}_{\mathrm{CDI}}^{[\mathrm{ext}]}(\mathbf{b}) = \left(L_{\mathrm{CDI}}^{[\mathrm{ext}]}(b_1), \ldots, L_{\mathrm{CDI}}^{[\mathrm{ext}]}(b_m), \ldots, L_{\mathrm{CDI}}^{[\mathrm{ext}]}(b_{\ell_b})\right),$$
$$\mathbf{L}_{\mathrm{CDII}}^{[\mathrm{ext}]}(\tilde{\mathbf{b}}) = \left(L_{\mathrm{CDII}}^{[\mathrm{ext}]}(\tilde{b}_1), \ldots, L_{\mathrm{CDII}}^{[\mathrm{ext}]}(\tilde{b}_m), \ldots, L_{\mathrm{CDII}}^{[\mathrm{ext}]}(\tilde{b}_{\ell_b})\right)$$

on the information bits $\mathbf{b}$ based on the LogMAP decoding algorithm [BCJR74, RVH95] (see Appendix C). Each convolutional SISO decoder receives three input LLR vectors: First, the channel-related LLRs

$$\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{b}) = \left(L_{\mathrm{DM}}^{[\mathrm{chan}]}(b_1), \ldots, L_{\mathrm{DM}}^{[\mathrm{chan}]}(b_m), \ldots, L_{\mathrm{DM}}^{[\mathrm{chan}]}(b_{\ell_b})\right) \quad \text{(Decoder I)},$$
$$\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\tilde{\mathbf{b}}) = \left(L_{\mathrm{DM}}^{[\mathrm{chan}]}(\tilde{b}_1), \ldots, L_{\mathrm{DM}}^{[\mathrm{chan}]}(\tilde{b}_m), \ldots, L_{\mathrm{DM}}^{[\mathrm{chan}]}(\tilde{b}_{\ell_b})\right) \quad \text{(Decoder II)}$$

on the systematic bits $\mathbf{b}$ and their interleaved version $\tilde{\mathbf{b}}$, respectively. Second, the channel-related LLRs

$$\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{c}^I) = \left(L_{\mathrm{DM}}^{[\mathrm{chan}]}(c_1^I), \ldots, L_{\mathrm{DM}}^{[\mathrm{chan}]}(c_i^I), \ldots, L_{\mathrm{DM}}^{[\mathrm{chan}]}(c_{\ell_c}^I)\right) \quad \text{(Decoder I)},$$

$$\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{c}^{II}) = \left(L_{\mathrm{DM}}^{[\mathrm{chan}]}(c_1^{II}), \ldots, L_{\mathrm{DM}}^{[\mathrm{chan}]}(c_i^{II}), \ldots, L_{\mathrm{DM}}^{[\mathrm{chan}]}(c_{\ell_c}^{II})\right) \quad \text{(Decoder II)}$$

on the respective parity bits as well as the *extrinsic* LLRs $\mathbf{L}_{\mathrm{CDII}}^{[\mathrm{ext}]}(\tilde{\mathbf{b}})$ or $\mathbf{L}_{\mathrm{CDI}}^{[\mathrm{ext}]}(\mathbf{b})$ on the information bits $\mathbf{b}$ as additional *a priori* information

$$\mathbf{L}_{\mathrm{CDI}}^{[\mathrm{apri}]}(\mathbf{b}) = \left(L_{\mathrm{CDI}}^{[\mathrm{apri}]}(b_1), \ldots, L_{\mathrm{CDI}}^{[\mathrm{apri}]}(b_m), \ldots, L_{\mathrm{CDI}}^{[\mathrm{apri}]}(b_{\ell_b})\right),$$

$$\mathbf{L}_{\mathrm{CDII}}^{[\mathrm{apri}]}(\tilde{\mathbf{b}}) = \left(L_{\mathrm{CDII}}^{[\mathrm{apri}]}(\tilde{b}_1), \ldots, L_{\mathrm{CDII}}^{[\mathrm{apri}]}(\tilde{b}_m), \ldots, L_{\mathrm{CDII}}^{[\mathrm{apri}]}(\tilde{b}_{\ell_b})\right).$$

After a certain number of iterations, MAP estimation is carried out based on the *a posteriori* LLRs $\mathbf{L}_{\mathrm{CDII}}(\mathbf{b}) = (L_{\mathrm{CDII}}(b_1), \ldots, L_{\mathrm{CDII}}(b_m), \ldots, L_{\mathrm{CDII}}(b_{\ell_b}))$ according to

$$\hat{\mathbf{b}} = \mathrm{sign}\{\mathbf{L}_{\mathrm{CDII}}(\mathbf{b})\} \quad \text{with} \quad \hat{\mathbf{b}} \in \{-1, 1\}^{\ell_b} \text{ (bipolar representation)}, \tag{2.5}$$

$$\mathbf{L}_{\mathrm{CDII}}(\mathbf{b}) = \mathbf{L}_{\mathrm{CDI}}^{[\mathrm{ext}]}(\mathbf{b}) + \mathbf{L}_{\mathrm{CDII}}^{[\mathrm{ext}]}(\mathbf{b}) + \mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{b}). \tag{2.6}$$

As an example, the *Bit Error Rate* (BER) of the rate-1/3 LTE Turbo code is depicted in Figure 2.3 for different numbers of employed Turbo iterations. This code consists of a parallel concatenation of two rate-1/2 convolutional codes with octal generator polynomials $\mathbf{G}_{\mathrm{CC}} = (1, 15/13)_8$ and constraint length $\mathfrak{L} + 1 = 4$, where $\mathfrak{L}$ denotes the number of delay elements. The simulation has been conducted in an AWGN environment using *Binary Phase Shift Keying* (BPSK) as

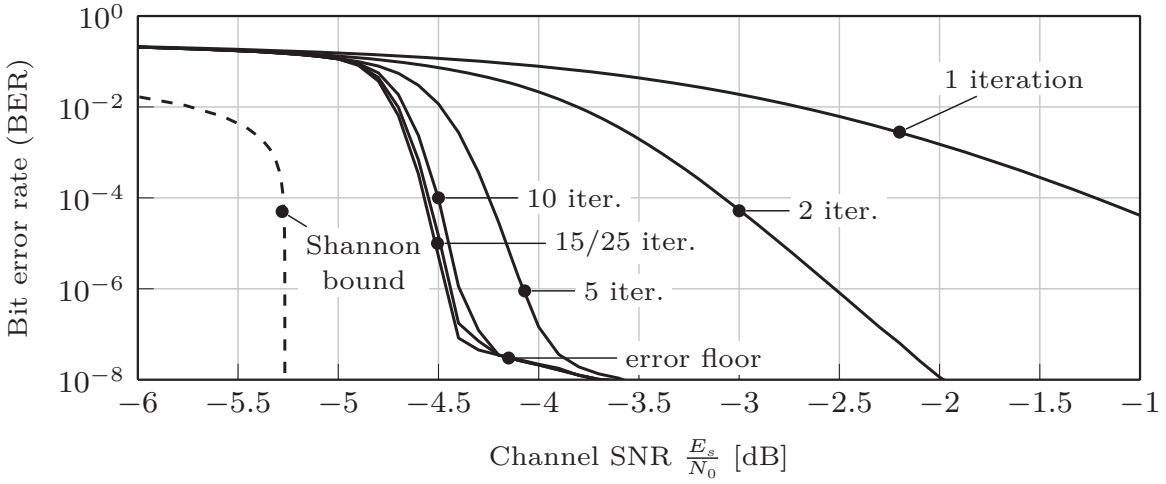


Figure 2.3: Bit error rate performance of the rate-1/3 LTE Turbo code with octal generator polynomials $\mathbf{G}_{\mathrm{CC}} = (1, 15/13)_8$ and constraint length $\mathfrak{L} + 1 = 4$ for a frame size of $\ell_b = 10000$ bits.

modulation scheme. The frame size has been set to $\ell_b = 10000$ bits. Furthermore, an S-random interleaver ($S = 15$) [DP95] has been used. This interleaver guarantees that no input bits within a distance S appear within a distance S at the output. Additionally, the Shannon bound for BPSK transmission [Moo05] (see Appendix A) is plotted as a reference which signifies the theoretical performance limit for infinite block lengths. The considered Turbo code is nearly converged after 10 to 15 Turbo iterations and shows the typical behavior of a PCCC, i.e., a steep waterfall and an error floor which is indicated in Figure 2.3. Furthermore, the Turbo code approaches the Shannon bound by approximately 0.8 dB for a target BER of 10^{-6}.

Serially Concatenated Convolutional Codes

The block diagram of a serially concatenated convolutional code is shown in Figure 2.4. The outer convolutional encoder of rate r_{out} obtains the information bits $\mathbf{b}$ and generates the encoded output vector $\mathbf{e} = (e_1, \ldots, e_i, \ldots, e_{\ell_e})$. These bits are then interleaved to the vector $\tilde{\mathbf{e}}$ and fed into the inner convolutional encoder. The inner convolutional encoder should be of rate $r_{\text{in}} \geq 1$ which is a necessary condition for capacity approaching systems [AKtB02, AKtB04]. Artificial redundancy is, thus, only inserted by the outer code. After bipolar transmission, channel-related LLRs $\mathbf{L}_{\text{DM}}^{[\text{chan}]}(\mathbf{y})$ are computed by means of (2.1) and provided to the inner convolutional SISO decoder. In accordance to the PCCC

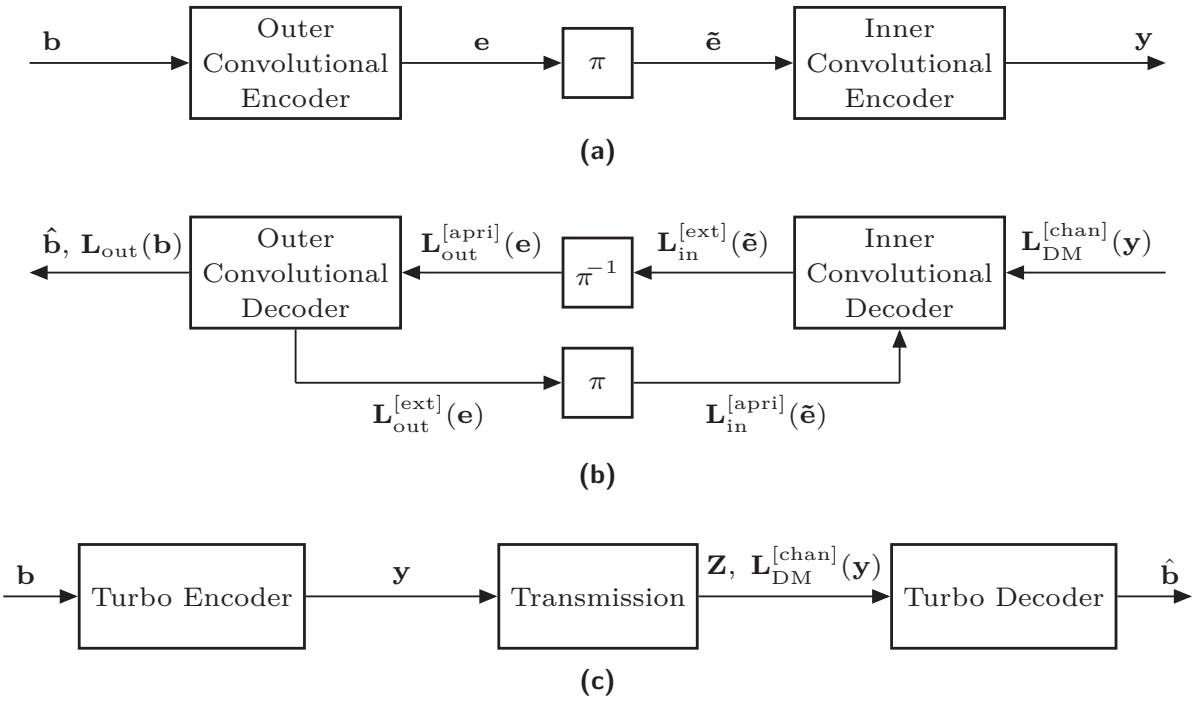


Figure 2.4: Turbo encoder (a) and Turbo decoder (b) for a serially concatenated convolutional code. A simplified transmission chain is shown in (c).

case, *extrinsic* LLRs $\mathbf{L}_{\text{in}}^{[\text{ext}]}(\tilde{\mathbf{e}}) = (L_{\text{in}}^{[\text{ext}]}(\tilde{e}_1), \ldots, L_{\text{in}}^{[\text{ext}]}(\tilde{e}_{\ell_e}))$ on the input bits $\tilde{\mathbf{e}}$ are calculated, deinterleaved, and forwarded to the outer convolutional SISO decoder as *a priori* information $\mathbf{L}_{\text{out}}^{[\text{apri}]}(\mathbf{e}) = (L_{\text{out}}^{[\text{apri}]}(e_1), \ldots, L_{\text{out}}^{[\text{apri}]}(e_{\ell_e}))$. Based on this information, the outer SISO decoder feeds back *extrinsic* information $\mathbf{L}_{\text{out}}^{[\text{ext}]}(\mathbf{e}) = (L_{\text{out}}^{[\text{ext}]}(e_1), \ldots, L_{\text{out}}^{[\text{ext}]}(e_{\ell_e}))$ on its output bits $\mathbf{e}$ starting the iterative loop. After a certain number of iterations, MAP estimation is carried out based on the *a posteriori* LLRs $\mathbf{L}_{\text{out}}(\mathbf{b}) = (L_{\text{out}}(b_1), \ldots, L_{\text{out}}(b_m), \ldots, L_{\text{out}}(b_{\ell_b}))$ according to

$$\hat{\mathbf{b}} = \text{sign}\{\mathbf{L}_{\text{out}}(\mathbf{b})\} \quad \text{with} \quad \hat{\mathbf{b}} \in \{-1, 1\}^{\ell_b} \text{ (bipolar representation)}, \tag{2.7}$$

$$\mathbf{L}_{\text{out}}(\mathbf{b}) = \mathbf{L}_{\text{out}}^{[\text{ext}]}(\mathbf{b}) + \mathbf{L}_{\text{in}}^{[\text{ext}]}(\mathbf{b}) + \mathbf{L}_{\text{DM}}^{[\text{chan}]}(\mathbf{b}). \tag{2.8}$$

Note that according to (2.8), the channel-related reliability values $\mathbf{L}_{\text{DM}}^{[\text{chan}]}(\mathbf{b})$ have to be forwarded to the outer decoder. However, this connection is omitted in Figure 2.4(b) for the sake of simplicity.

The BER of the rate-1/2 serially concatenated convolutional code presented in [tB00a] is depicted in Figure 2.5. The Shannon bound for BPSK transmission is illustrated as reference again. The Turbo code consists of a rate-1 inner code with octal generator polynomial $G_{\text{in}} = (2/3)_8$ and constraint length $\mathfrak{L} + 1 = 2$ as well as a rate-1/2 outer code with octal generator polynomials $\mathbf{G}_{\text{out}} = (1, 5/7)_8$ and constraint length $\mathfrak{L} + 1 = 3$. The frame size has been set to $\ell_b = 10000$ bits and an S-random interleaver with $S = 15$ has been used again. As shown in Figure 2.5, the decoding process converges after approximately 15 iterations. Furthermore,

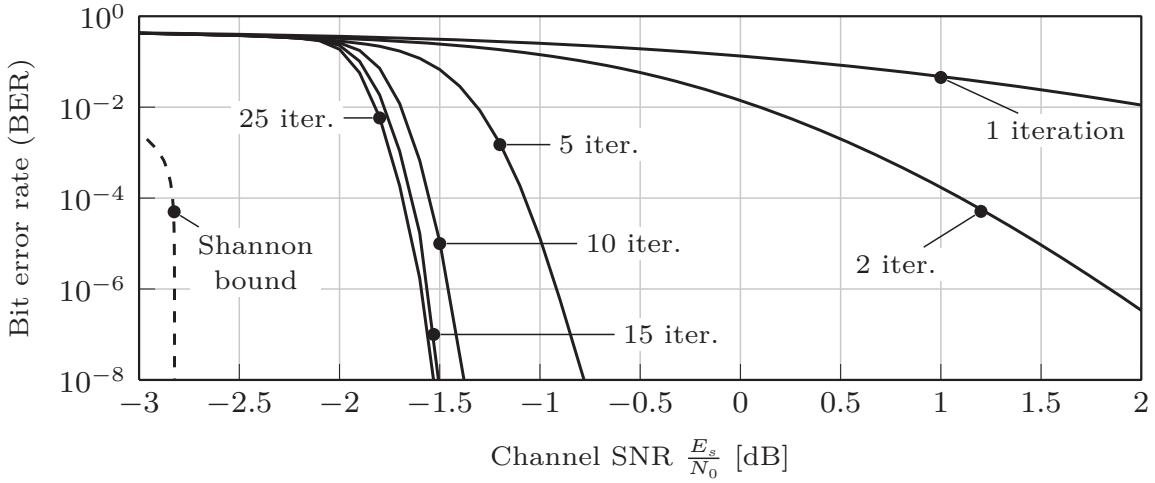


Figure 2.5: Bit error rate performance of the serially concatenated rate-1/2 Turbo code proposed in [tB00a] consisting of a rate-1 inner code with octal generator polynomial $G_{\text{in}} = (2/3)_8$ and constraint length $\mathfrak{L} + 1 = 2$ as well as a rate-1/2 outer code with octal generator polynomials $\mathbf{G}_{\text{out}} = (1, 5/7)_8$ and constraint length $\mathfrak{L} + 1 = 3$. Frame size: $\ell_b = 10000$ bits.

no relevant error floor is visible for the fully converged Turbo code since the BER falls rapidly below 10^{-8}. This is one major difference to PCCCs. The gap to the theoretical Shannon bound at a BER of 10^{-6} amounts to approximately 1.2 dB and is, thus, 0.4 dB higher than the gap of the converged LTE Turbo code.

2.2.2 Low Density Parity Check Codes

LDPC codes have originally been invented by Gallager in 1962 [Gal62, Gal63] and rediscovered by MacKay in 1997 [Mac97, Mac99]. They provide decoding performance comparable to that of Turbo codes. Sophisticated iterative decoding algorithms have been developed which enable a near-optimal decoding quality close to the Shannon limit with a decoding complexity per iteration much lower than the per-iteration complexity of common Turbo decoders. However, the total decoding complexity per block is of the same magnitude since considerably more iterations have to be carried out during the decoding process. Nevertheless, LDPC codes have several advantages compared to Turbo codes which have led to their inclusion in many standard as, e.g., WLAN [IEE09], WiMAX [IEE04], DVB-S2 [ETS09a] and DVB-T2 [ETS09b]. A high advantage of LDPC codes is that they surpass Turbo codes in terms of decoding performance for high code rates. Furthermore, a stopping criterion for reliable decoding is inherently given by the check equations of an LDPC code, while for Turbo codes such a check has conventionally to be performed by means of a CRC necessitating additional code rate and decoding complexity. Finally, they can efficiently be implemented in hardware, since many operations can be done in parallel.

LDPC codes are a class of linear block codes which are characterized by so-called sparse[1] binary[2] parity-check matrices $\mathbf{A} \in \mathbb{F}_2^{\ell_c \times \ell_y}$ of dimension $\ell_c \times \ell_y$, with ℓ_c denoting the number of parity bits, ℓ_y the total number of coded bits and $\ell_b = \ell_y - \ell_c$ the number of information bits. The code rate is then given by

$$r_{\mathrm{LDPC}} = \frac{\ell_y - \mathrm{rank}(\mathbf{A})}{\ell_y}, \tag{2.9}$$

where $\mathrm{rank}(\mathbf{A})$ denotes the rank of the parity-check matrix $\mathbf{A}$. If it has full rank, i.e., $\mathrm{rank}(\mathbf{A}) = \ell_c$, (2.9) simplifies to

$$r_{\mathrm{LDPC}} = \frac{\ell_y - \ell_c}{\ell_y} = \frac{\ell_b}{\ell_y}. \tag{2.10}$$

Since LDPC codes are linear block codes, they can be expressed as the null

[1] The term sparse means that the number of zero entries is high compared to the number of one entries.

[2] Only binary LDPC codes are considered here. However, LDPC codes can also be constructed over higher Galois fields [DM98].

space of $\mathbf{A}$ [RU01b]:

$$\mathbf{A} \cdot \mathbf{y}^T = \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_\nu \\ \vdots \\ \mathbf{A}_{\ell_c} \end{pmatrix} \cdot \mathbf{y}^T = \mathbf{0}, \quad \begin{matrix} \mathbf{A}_\nu = (A_{\nu,1}, \ldots, A_{\nu,n}, \ldots, A_{\nu,\ell_y}) \\ \mathbf{y} = (y_1, \ldots, y_n, \ldots, y_{\ell_y}) \end{matrix} . \quad (2.11)$$

The equation $\mathbf{A}_\nu \cdot \mathbf{y}^T = 0$ is called the ν-th check equation regarding the codeword $\mathbf{y}$. Following the description in [Moo05], two special sets can now be defined:

(a) $\mathfrak{N}_\nu = \{n : A_{\nu,n} = 1\}$ contains all bits that participate in parity check equation ν, $1 \leq \nu \leq \ell_c$.

(b) $\mathfrak{I}_n = \{\nu : A_{\nu,n} = 1\}$ denotes the set of all check equations in which bit n participates.

The cardinalities $|\mathfrak{N}_\nu|$ and $|\mathfrak{I}_n|$ are the weight of row ν and column n, respectively. If all row weights as well as all column weights are identical, the LDPC code is called *regular*, otherwise *irregular* [LMSS01,RU08]. A regular code is, thus, defined by the constant column weight w_v (variable node degree) and row weight w_c (check node degree) and denoted as $(w_\mathrm{v}, w_\mathrm{c})$-regular LDPC code.

Irregular LDPC codes are characterized by a so-called *degree distribution pair* $(\lambda(\chi), \varrho(\chi))$ which can either be given in the *node perspective* or in the *edge perspective*. The polynomials

$$\lambda(\chi) = \sum_{i=1}^{w_{\mathrm{v,max}}} \lambda_i \cdot \chi^i \quad \text{and} \quad \varrho(\chi) = \sum_{j=1}^{w_{\mathrm{c,max}}} \varrho_j \cdot \chi^j \quad (2.12)$$

represent the node perspective with $w_{\mathrm{v,max}}$ and $w_{\mathrm{c,max}}$ signifying the maximum variable and check node degree. The coefficients λ_i and ϱ_j denote the fractions of nodes with degree i and j, respectively. The edge perspective describes the probability that a certain edge is connected to a variable node of degree i and to a check node of degree j, respectively:

$$\lambda'(\chi) = \frac{\frac{\mathrm{d}}{\mathrm{d}\chi}\lambda(\chi)}{\frac{\mathrm{d}}{\mathrm{d}\chi}\lambda(\chi)|_{\chi=1}} = \sum_{i=2}^{w_{\mathrm{v,max}}} \lambda'_i \cdot \chi^{i-1}, \quad (2.13)$$

$$\varrho'(\chi) = \frac{\frac{\mathrm{d}}{\mathrm{d}\chi}\varrho(\chi)}{\frac{\mathrm{d}}{\mathrm{d}\chi}\varrho(\chi)|_{\chi=1}} = \sum_{j=2}^{w_{\mathrm{c,max}}} \varrho'_j \cdot \chi^{j-1}. \quad (2.14)$$

Irregular LDPC codes provide potentially better performance within the waterfall region than regular LDPC codes [LMSS01, RU01b], however, they suffer from an

increased error floor [MWD99]. A number of good irregular rate-1/2 LDPC codes are summarized in [Moo05].

An LDPC code is commonly described by a bipartite graph consisting of ℓ_y variable nodes and ℓ_c check nodes (known as factor or Tanner graph [Tan81]). Each variable node corresponds to a code bit and each check node corresponds to a parity check equation as defined by a row of the parity check matrix. Those variable nodes corresponding to information bits are denoted as *information nodes*, while variable nodes corresponding to parity bits are called *parity nodes*. Variable node n is connected to all check nodes in the set $\mathfrak{I}_n$ and check node ν is connected to all variable nodes in the set $\mathfrak{N}_\nu$. As an example, the Tanner graph of the rate-1/2 irregular LDPC code with parity check matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \tag{2.15}$$

is shown in Figure 2.6. The variable nodes are connected to the check nodes according to the rows of the parity check matrix.

Optimal MAP decoding is unfeasible due to complexity constraints. However, LDPC codes can efficiently be decoded by *message passing algorithms* as already proposed by Gallager [Gal62]. For soft decoding of LDPC codes, the iterative Belief Propagation algorithm as described in [Pea88] is commonly applied. Messages in terms of probabilities or LLRs are passed across the edges of the Tanner graph. If there do not exist cycles within the Tanner graph, the incoming messages at each node are statistically independent resulting in optimal MAP decoding [Pea88]. However, a Tanner graph is not free of cycles in general. Therefore, the *a posteriori* probabilities obtained at the end of the iteration process are only approximate solutions, since the information exchanged along the edges is not statistically in-

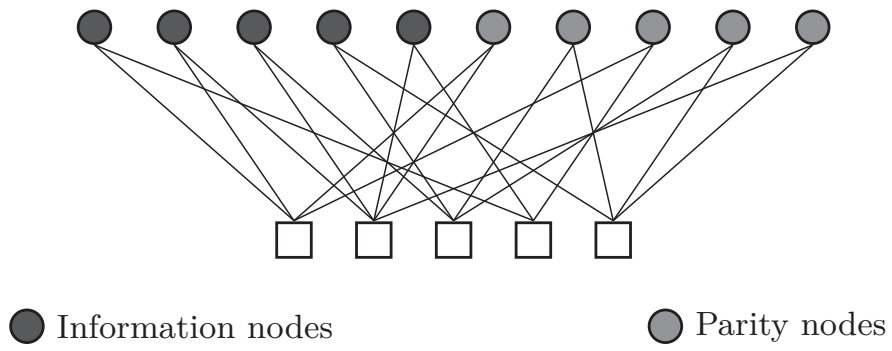


Figure 2.6: Exemplary Tanner graph for the rate-1/2 irregular LDPC code given by (2.15) and a block length of $\ell_b = 5$.

dependent anymore.

The Belief Propagation algorithm employed throughout this thesis can be summarized as follows: After initialization of each variable-to-check message $\mathfrak{v}_{\nu,n}$ (i.e., the message sent from variable node n to check node ν) with the received channel-related LLR $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n)$ for the n-th bit, the so called horizontal and vertical step are computed alternately. The horizontal step (or check node update) computes the check-to-variable messages $\mathfrak{c}_{\nu,n}$ (i.e., the message sent from check node ν to variable node n) as

$$\mathfrak{c}_{\nu,n} = 2\tanh^{-1}\left(\prod_{n'\in\mathfrak{N}_\nu\setminus n}\tanh\left(\frac{\mathfrak{v}_{\nu,n'}}{2}\right)\right) = \boxplus\!\!\sum_{n'\in\mathfrak{N}_\nu\setminus n}\mathfrak{v}_{\nu,n'}, \tag{2.16}$$

where the operator "\" denotes exclusion of an element from a set and $\boxplus$ signifies the box-plus operation [HOP96]. In practice, either lookup tables for the tanh-function and its inverse or approximations are used for the computation of the check node update in order to limit the computational effort. The most famous approximation has been proposed in [HOP96]:

$$\mathfrak{c}_{\nu,n} \approx \left(\prod_{n'\in\mathfrak{N}_\nu\setminus n}\operatorname{sign}\left(\mathfrak{v}_{\nu,n'}\right)\right)\cdot\min_{n'\in\mathfrak{N}_\nu\setminus n}\left|\mathfrak{v}_{\nu,n'}\right|. \tag{2.17}$$

Likewise, the vertical step updates all variable-to-check messages according to

$$\mathfrak{v}_{\nu,n} = \mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n) + \sum_{\nu'\in\mathfrak{I}_n\setminus\nu}\mathfrak{c}_{\nu',n}\,. \tag{2.18}$$

Additionally, after each vertical step, the hard decision

$$\hat{y}_n = \operatorname{sign}\left(\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n) + \sum_{\nu'\in\mathfrak{I}_n}\mathfrak{c}_{\nu',n}\right) = \operatorname{sign}\left(L_{\mathrm{VN}}(y_n)\right) \tag{2.19}$$

is computed for each variable node n, based on its *a posteriori* LLR $L_{\mathrm{VN}}(y_n)$, to evaluate the parity check equations. Decoding is stopped if either all parity check equations are fulfilled or the maximum number of iterations is reached. For more details on LDPC codes, the reader is referred to [Moo05, RU08].

The bit error rate performance of an rate-1/3 irregular LDPC code with degree distributions (node perspective) $\lambda(\chi) = 0.565771\cdot\chi^2 + 0.211528\cdot\chi^3 + 0.118725\cdot\chi^4 + 0.103976\cdot\chi^{11}$ and $\varrho(\chi) = 0.922857\cdot\chi^5 + 0.0771428\cdot\chi^6$ is illustrated in Figure 2.7 dependent on the number of iterations. The degree distributions have been optimized with density evolution [RSU01] and the *Progressive Edge Growth* (PEG) algorithm proposed by Hu et al. [HEA05] has been used to generate the LDPC code. The frame size has been set to $\ell_b = 10000$ bits in this case. In addition

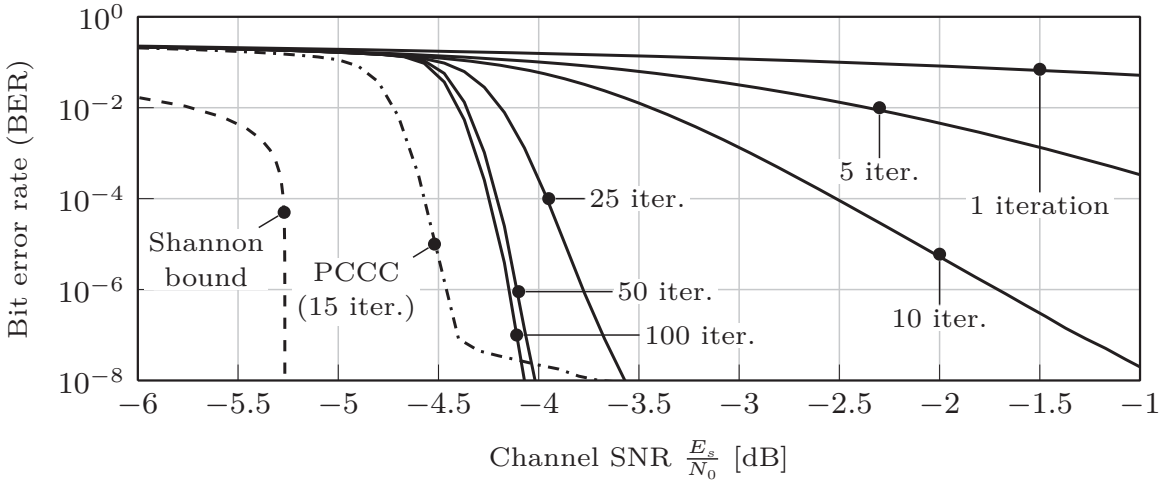


Figure 2.7: Bit error rate performance of a rate-1/3 PEG LDPC code (ℓ_b = 10000 bits).

to the BER curves of the LDPC code, the Shannon bound for BPSK transmission (dashed line) as well as the PCCC employed in LTE (dash dotted line) for 15 Turbo iterations are depicted. The LDPC code converges after approximately 50 iterations and shows no considerable error floor in contrast to the PCCC. However, there is a gap of 0.3 dB in terms of channel SNR to the PCCC within the waterfall region resulting in a gap of 1.1 dB to the Shannon bound.

2.2.3 Convergence Analysis Using EXIT Charts

The invention of *EXtrinsic Information Transfer* (EXIT) charts based on the concept of mutual information by ten Brink in [tB00b, tB01a] is seen as breakthrough in the area of code design for iteratively decoded coding schemes. EXIT charts describe the flow of *extrinsic* information through the constituent SISO decoders and are a powerful tool to predict the convergence behavior of the iterative decoder, particularly within the waterfall region. It is well known for LDPC codes that EXIT charts provide only a poor prediction of the convergence behavior at low channel qualities. Therefore, these codes are commonly designed and/or analyzed based on density evolution [RU01a] which determines an SNR threshold above which perfect decoding becomes (theoretically) possible. In the case of Turbo codes, however, EXIT charts are a widely-used analysis tool. The following description is, thus, restricted to Turbo codes. An adaptation to LDPC codes is, e.g., given in [AKtB02, AKtB04].

Parallel Concatenation

A general information theoretic transmission model has been presented in [AKtB02, AKtB04] which enables the evaluation of PCCCs as well as SCCCs. For the sake of clarity, both concatenation schemes are individually covered by simplified versions

of the general model. The simplified model which corresponds to the PCCC case is depicted in Figure 2.8. For an information theoretic evaluation, each information bit b_m and code bit y_n can be interpreted as realization of a random variable B_m and Y_n, respectively. Accordingly, each *a priori* LLR $L_{\mathrm{CD}}^{[\mathrm{apri}]}(b_m)$ and communication channel-related LLR[3] $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n)$ can be modeled by random variables $\mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}]}(b_m)$ and $\mathcal{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n)$.

The SISO decoder receives two types of input vectors: The first vector contains the LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{y})$ of the code bits $\mathbf{y}$ which are delivered by the communication channel. This channel comprises, besides the model for the "real" communication channel, the mapper and soft demapper as well. The second vector contains the *extrinsic* LLRs of the information bits $\mathbf{b}$ provided by the other component decoder of the PCCC which is exploited by the SISO decoder as *a priori* information $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}]}(\mathbf{b})$. The feedback of information is modeled by a so-called *extrinsic* channel which is an artificial device that is beneficial for the theoretical evaluation but does not exist in reality. Both types of information are then exploited in order to determine the *extrinsic* LLRs $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{ext}]}(\mathbf{b})$ as well as the *a posteriori* LLRs $\mathbf{L}_{\mathrm{CD}}(\mathbf{b})$ of the information bits $\mathbf{b}$ according to (2.6).

In this model, the communication channel is characterized by the average mutual information between its input bits $\mathbf{y}$ and its output LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{y})$:

$$\mathcal{I}_{\mathrm{C}} = \frac{1}{\ell_y} \sum_{n=1}^{\ell_y} \mathcal{I}\left(\mathcal{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n); Y_n\right). \tag{2.20}$$

Similarly, the *extrinsic* channel is characterized by the average mutual information between its input bits $\mathbf{b}$ and its output LLRs $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}]}(\mathbf{b})$:

$$\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} = \frac{1}{\ell_b} \sum_{m=1}^{\ell_b} \mathcal{I}\left(\mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}]}(b_m); B_m\right). \tag{2.21}$$

[3] In the following, these LLRs are denoted as *channel-related LLRs* omitting the term *communication*.

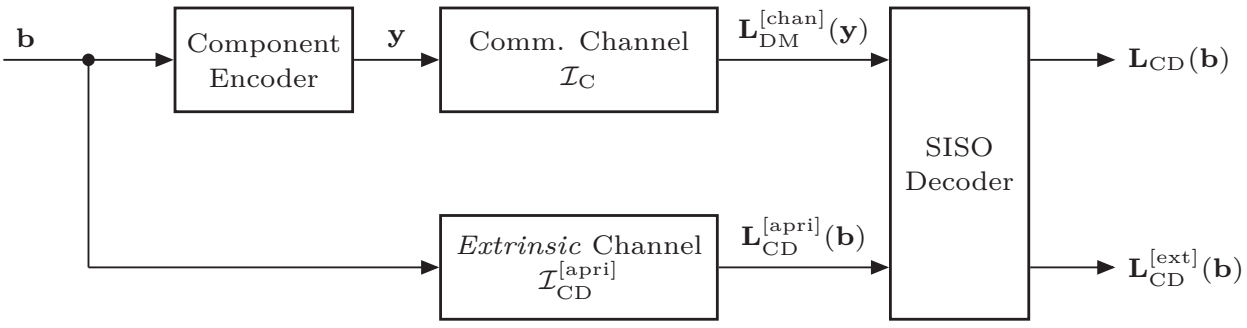


Figure 2.8: General information theoretic transmission model for component codes of PCCCs.

The EXIT characteristic $\mathcal{T}_{\mathrm{CD}}$ is now defined by means of the *a priori* information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}$ and the channel-related mutual information $\mathcal{I}_{\mathrm{C}}$ according to

$$\mathcal{T}_{\mathrm{CD}}\left(\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}\middle|\,\mathcal{I}_{\mathrm{C}}\right)=\mathcal{I}_{\mathrm{CD}}^{[\mathrm{ext}]}=\frac{1}{\ell_b}\sum_{m=1}^{\ell_b}\mathcal{I}\left(\mathcal{L}_{\mathrm{CD}}^{[\mathrm{ext}]}(b_m);B_m\right) \tag{2.22}$$

where $\mathcal{I}_{\mathrm{C}}$ is considered as arbitrary but fixed. The EXIT characteristic illustrates the amount of *extrinsic* information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{ext}]}$ which is generated by a SISO decoder dependent on the amount of *a priori* information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}$ at a specific channel quality (relates to $\mathcal{I}_{\mathrm{C}}$).

Serial Concatenation

Serially concatenated codes consist of an inner code and an outer code as illustrated in Figure 2.4(a). The inner decoder determines *extrinsic* LLRs $\mathbf{L}_{\mathrm{in}}^{[\mathrm{ext}]}(\tilde{\mathbf{e}})$ of the input bits $\tilde{\mathbf{e}}$ of the inner encoder, which is identical to the PCCC case. The definition of the EXIT characteristic $\mathcal{T}_{\mathrm{in}}$ is, thus, given by adopting the notation accordingly:

$$\mathcal{T}_{\mathrm{in}}\left(\mathcal{I}_{\mathrm{in}}^{[\mathrm{apri}]}\middle|\,\mathcal{I}_{\mathrm{C}}\right)=\mathcal{I}_{\mathrm{in}}^{[\mathrm{ext}]}=\frac{1}{\ell_e}\sum_{i=1}^{\ell_e}\mathcal{I}\left(\mathcal{L}_{\mathrm{in}}^{[\mathrm{ext}]}(\tilde{e}_i);\tilde{\mathcal{E}}_i\right), \tag{2.23}$$

$$\mathcal{I}_{\mathrm{in}}^{[\mathrm{apri}]}=\frac{1}{\ell_e}\sum_{i=1}^{\ell_e}\mathcal{I}\left(\mathcal{L}_{\mathrm{in}}^{[\mathrm{apri}]}(\tilde{e}_i);\tilde{\mathcal{E}}_i\right), \tag{2.24}$$

$$\mathcal{I}_{\mathrm{C}}=\frac{1}{\ell_y}\sum_{n=1}^{\ell_y}\mathcal{I}\left(\mathcal{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n);Y_n\right). \tag{2.25}$$

The variables $\tilde{\mathcal{E}}_i$, $\mathcal{L}_{\mathrm{in}}^{[\mathrm{ext}]}(\tilde{e}_i)$, Y_n, and $\mathcal{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(y_i)$ signify the corresponding random processes.

However, the outer decoder computes *extrinsic* LLRs $\mathbf{L}_{\mathrm{out}}^{[\mathrm{ext}]}(\mathbf{e})$ of the output bits $\mathbf{e}$ and not regarding the input bits $\mathbf{b}$ of the outer encoder. The corresponding information theoretic model is depicted in Figure 2.9. Accordingly, the outer SISO decoder only exploits the *a priori* LLRs $\mathbf{L}_{\mathrm{out}}^{[\mathrm{apri}]}(\mathbf{e})$ which are provided by the

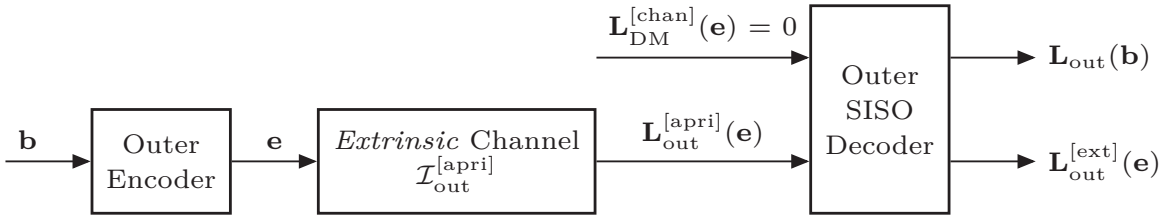


Figure 2.9: General information theoretic transmission model for outer codes of SCCCs.

inner SISO decoder. No channel-related LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{e})$ are given to compute the *extrinsic* LLRs $\mathbf{L}_{\mathrm{out}}^{[\mathrm{ext}]}(\mathbf{e})$ and the *a posteriori* LLRs $\mathbf{L}_{\mathrm{out}}(\mathbf{b})$, since non-systematic recursive convolutional inner codes with $r_{\mathrm{in}} = 1$ are commonly employed. Similar to the PCCC case, the average *a priori* information is, thus, defined as

$$\mathcal{I}_{\mathrm{out}}^{[\mathrm{apri}]} = \frac{1}{\ell_e} \sum_{i=1}^{\ell_e} \mathcal{I}\left(\mathcal{L}_{\mathrm{out}}^{[\mathrm{apri}]}(e_i); \mathcal{E}_i\right), \tag{2.26}$$

where $\mathcal{L}_{\mathrm{out}}^{[\mathrm{apri}]}(e_i)$ and $\mathcal{E}_i$ are random variables with realization $L_{\mathrm{out}}^{[\mathrm{apri}]}(e_i)$ and e_i $(i = 1 \ldots \ell_e)$, respectively.

The EXIT characteristic $\mathcal{T}_{\mathrm{out}}$ is then defined by means of the *a priori* information $\mathcal{I}_{\mathrm{out}}^{[\mathrm{apri}]}$ according to

$$\mathcal{T}_{\mathrm{out}}\left(\mathcal{I}_{\mathrm{out}}^{[\mathrm{apri}]}\right) = \mathcal{I}_{\mathrm{out}}^{[\mathrm{ext}]} = \frac{1}{\ell_e} \sum_{i=1}^{\ell_e} \mathcal{I}\left(\mathcal{L}_{\mathrm{out}}^{[\mathrm{ext}]}(e_i); \mathcal{E}_i\right). \tag{2.27}$$

Simulating EXIT charts

In order to obtain the desired EXIT characteristics, Monte-Carlo simulations have to be carried out in general. An analytical derivation of the *extrinsic* information is only possible for special codes as, e.g., repetition codes or special LDPC codes and/or special channel models as, e.g., the BEC and the BSC [AKtB04, LHHH05].

However, the *a priori* information can be determined analytically for various types of *extrinsic* channels. Throughout this contribution the major information theoretic considerations are restricted to the BEC and the *Binary-Input Additive White Gaussian Noise Channel* (BIAWGNC). Nevertheless, similar derivations can be carried out, e.g., for the BSC [CT91, LHHH05].

As the following considerations are identical for PCCCs and SCCCs, all subscripts and superscripts associated with the respective concatenation scheme are skipped unless there is a risk of confusion. The analysis is performed exemplarily for the PCCC case. A transformation to the SCCC case is straightforward by substituting the vector $\mathbf{b}$ with $\mathbf{e}$. Furthermore, the sums and the normalization factors in (2.21) and (2.26) can be omitted, since all information bits and their related *a priori* LLRs are i.i.d.. Consequently, both vectors can be interpreted as realizations of the same random variables B and $\mathcal{L}^{[\mathrm{apri}]}(b)$, respectively. Therefore, also the subscript m is skipped in what follows. A full derivation of the capacity terms are given in Appendix A.

BEC case: The channel capacity $C = \mathcal{I}^{[\mathrm{apri}]}$ is given by means of the erasure probability P_{E} [CT91]:

$$\mathcal{I}^{[\mathrm{apri}]} = 1 - P_{\mathrm{E}}. \tag{2.28}$$

BSC case: The channel capacity $C = \mathcal{I}^{[\text{apri}]}$ is dependent on the binary entropy function $H'(P_0)$ in which P_0 denotes the cross-over probability [CT91]:

$$\mathcal{I}^{[\text{apri}]} = 1 - H'(P_0) = 1 + P_0 \cdot \log_2(P_0) + (1 - P_0) \cdot \log_2(1 - P_0). \tag{2.29}$$

BIAWGNC case: Each *a priori* LLR $L^{[\text{apri}]}(b)$ can be modeled by an independent Gaussian random variable $\mathcal{L}_A = \mathcal{L}^{[\text{apri}]}(b)$ with realization $\xi = L^{[\text{apri}]}(b)$ and conditional probability density function

$$p_{\mathcal{L}_A|B}\left(\mathcal{L}_A = \xi | B = b\right) = \frac{1}{\sqrt{2\pi}\sigma_{\mathcal{L}}} \cdot \exp\left(-\frac{\left(\xi - \frac{\sigma_{\mathcal{L}}^2}{2} b\right)^2}{2\sigma_{\mathcal{L}}^2}\right), \tag{2.30}$$

where $\sigma_{\mathcal{L}}^2$ denotes the variance and $\mu_{\mathcal{L}} = \sigma_{\mathcal{L}}^2/2$ the mean of the probability distribution.

Hence, the mutual information $\mathcal{I}^{[\text{apri}]}$ between the transmitted bipolar information bits b and their corresponding *a priori* LLRs $\mathcal{L}^{[\text{apri}]}(b)$ can be expressed according to [CT91]:

$$\begin{aligned}\mathcal{I}^{[\text{apri}]} &= \mathcal{I}\left(\mathcal{L}_A = \xi; B = b\right) \\ &= \frac{1}{2} \sum_{b=\pm 1} \int_{-\infty}^{\infty} p_{\mathcal{L}_A|B}(\xi|b) \, \text{ld} \frac{2 \cdot p_{\mathcal{L}_A|B}(\xi|B=b)}{p_{\mathcal{L}_A|B}(\xi|B=-1) + p_{\mathcal{L}_A|B}(\xi|B=1)} d\xi. \end{aligned} \tag{2.31}$$

Substituting (2.30) into (2.31) results in ten Brink's well known J-function [tB01a]

$$J(\sigma_{\mathcal{L}}) = \mathcal{I}^{[\text{apri}]} = 1 - \int_{-\infty}^{\infty} \frac{\exp\left(-\frac{\left(\xi - \frac{\sigma_{\mathcal{L}}^2}{2}\right)^2}{2\sigma_{\mathcal{L}}^2}\right)}{\sqrt{2\pi}\sigma_{\mathcal{L}}} \log_2\left(1 + e^{-\xi}\right) d\xi, \tag{2.32}$$

which cannot be expressed in closed-form and, thus, has to be numerical approximated. This function provides an interrelation between $\mathcal{I}^{[\text{apri}]}$ and $\sigma_{\mathcal{L}}$. Consequently, Monte Carlo simulations can be carried out for approximating the flow of *extrinsic* information by simply varying $\sigma_{\mathcal{L}}$.

One can further conclude from (2.32) that the capacity of a BIAWGNC is given by

$$C_{\text{BIAWGNC}} = J\left(\sigma_{\mathcal{L}}\right), \quad \sigma_{\mathcal{L}} = 2/\sigma_n, \tag{2.33}$$

where $\sigma_n^2 = N_0/2$ is the variance of the Gaussian distributed noise signal and $N_0/2$ the spectral noise power density. For an energy per modulation symbol of $E_s = 1$,

the channel SNR E_s/N_0 can be expressed according to

$$\frac{E_s}{N_0} = \frac{1}{2\sigma_n^2} \quad \Leftrightarrow \quad \sigma_n = \frac{1}{\sqrt{2\frac{E_s}{N_0}}}. \tag{2.34}$$

The ratio E_b/N_0 of the energy per information bit and the spectral noise power density further incorporates the overall code rate r and the number of information bits I per modulation symbol:

$$\frac{E_b}{N_0} = \frac{1}{I \cdot r} \cdot \frac{E_s}{N_0}. \tag{2.35}$$

A comprehensive analysis of all channel models is given in Appendix A.

Considering a PCCC or an inner code of a SCCC scheme, channel-related LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{y})$ are also provided to the SISO decoder. In case of a BEC, BSC, or BIAWGNC[4], these LLRs can be generated for each channel quality in accordance to the above models. In general, however, Monte-Carlo simulations are required to determine the output LLRs of the communication channel with mutual information $\mathcal{I}_{\mathrm{C}}$. The *a priori* LLRs $\mathbf{L}^{[\mathrm{apri}]}(\mathbf{b})$, which are also generated by one of the above analyzed channel models, as well as the channel-related LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{y})$ are then exploited by the SISO decoder to determine the *extrinsic* LLRs $\mathbf{L}^{[\mathrm{ext}]}(\mathbf{b})$. Modeling these LLRs by a random process $\mathcal{L}_{\mathrm{E}} = \mathcal{L}^{[\mathrm{ext}]}(b)$ with realization $\xi = L^{[\mathrm{ext}]}(b)$, two measurements of the conditional output probability density functions $p_{\mathcal{L}_E|B}(\mathcal{L}_E = \xi|B = -1)$ and $p_{\mathcal{L}_E|B}(\mathcal{L}_E = \xi|B = 1)$ can finally be performed to numerically approximate the *extrinsic* information at the output of the SISO decoder:

$$\begin{aligned} \mathcal{I}^{[\mathrm{ext}]} &= \mathcal{I}\left(\mathcal{L}_E = \xi; B = b\right) \\ &= \frac{1}{2} \sum_{b=\pm 1} \int_{-\infty}^{\infty} p_{\mathcal{L}_E|B}(\xi|b) \, \mathrm{ld} \frac{2 \cdot p_{\mathcal{L}_E|B}(\xi|b)}{p_{\mathcal{L}_E|B}(\xi|B=-1) + p_{\mathcal{L}_E|B}(\xi|B=1)} d\xi. \end{aligned} \tag{2.36}$$

Examples

The EXIT charts of a PCCC and an SCCC are given in Figure 2.10 for distinct channel SNRs E_s/N_0 and BPSK modulation. As a representative of a PCCC, the systematic rate-1/3 LTE Turbo code with octal generator polynomials $\mathbf{G}_{\mathrm{CC}} = (1, 15/13)_8$ and constraint length $\mathfrak{L} + 1 = 4$ is considered at $E_s/N_0 = -4.5\,\mathrm{dB}$ in Figure 2.10(a). In an EXIT chart, the EXIT characteristics of both component codes are plotted together in one diagram. However, the EXIT characteristic of the second component has to be plotted with swapped axes, since it exploits the *extrinsic* information of the first component as *a priori* knowledge. The iterative

[4]Note that AWGN transmission with QPSK can be transformed to the BIAWGNC case, since BPSK demodulation is performed for each bit of the modulation symbol. This is the difference to higher modulation orders.

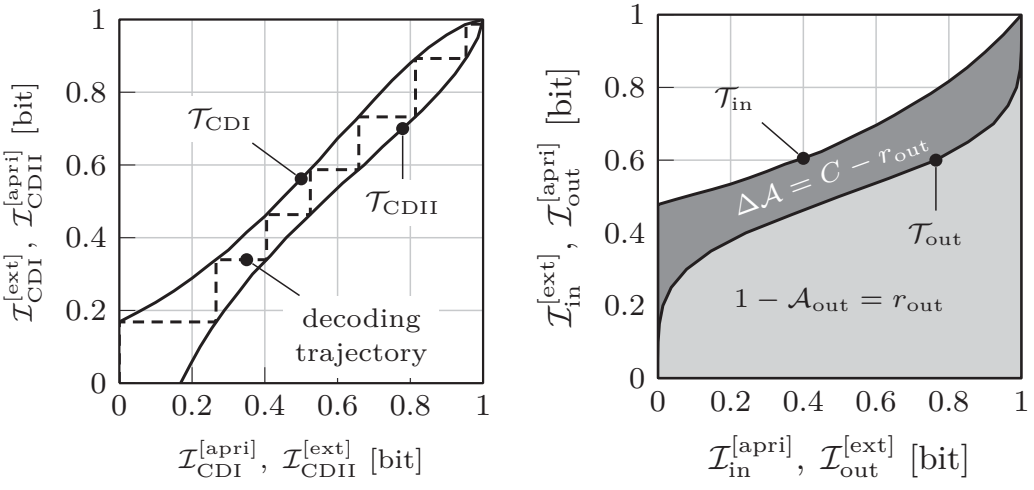
exchange of *extrinsic* information can then be visualized in Figure 2.10(a) by a so-called *decoding trajectory* (dashed line). Near error free decoding is possible if an open decoding tunnel exists, i.e., if both EXIT characteristics do not intersect. In this case, the point (1,1) (top right corner) in the EXIT chart is reached after a certain number of iterations which implies that perfect *extrinsic* information ($\mathcal{I}$=1 bit) can be generated. The openness of the tunnel and, consequently, the number of required iterations is influenced by the channel SNR E_s/N_0.

An example for an SCCC has been presented in [tB00a]. The inner code is a rate-1 *Recursive Non-Systematic Convolutional* (RNSC) code with octal generator polynomial $G_{\text{in}} = (2/3)_8$ and constraint length $\mathfrak{L} + 1 = 2$. The outer code is a rate-1/2 *Recursive Systematic Convolutional* (RSC) code with octal generator polynomials $\mathbf{G}_{\text{out}} = (1, 5/7)_8$ and constraint length $\mathfrak{L} + 1 = 3$. The EXIT chart is depicted in Figure 2.10(b) for $E_s/N_0 = -0.5\,\text{dB}$. A special EXIT chart property which is called the area property is visualized. This property can be subdivided into a capacity property and a rate property and has been derived in [AKtB04] and recalled, e.g., in [Hag04].

Capacity property: If the rate of the inner code is $r_{\text{in}} = 1$, then the area under the EXIT characteristic of the inner code is given by

$$\mathcal{A}_{\text{in}} = C, \tag{2.37}$$

where C denotes the channel capacity.



(a) PCCC:
$\mathbf{G}_{\text{CC}} = (1, 15/13)_8$
$E_s/N_0 = -4.5\,\text{dB}$

(b) SCCC:
$\mathbf{G}_{\text{in}} = (2/3)_8$
$\mathbf{G}_{\text{out}} = (1, 5/7)_8$
$E_s/N_0 = -0.5\,\text{dB}$

Figure 2.10: Properties of EXIT characteristics.

Rate property: The area under the swapped EXIT characteristic of the outer decoder amounts to

$$1 - \mathcal{A}_{\text{out}} = r_{\text{out}}, \tag{2.38}$$

where r_{out} is the code rate of the outer code.

Iterative decoding is now possible, if the area between both curves is larger than zero, i.e., $\Delta\mathcal{A} = \mathcal{A}_{\text{in}} - (1 - \mathcal{A}_{\text{out}}) = C - r_{\text{out}} > 0$ and if there is no intersection between both characteristics. Ashikhmin et al. have further shown that inner code rates of $r_{\text{in}} < 1$ cause an inherent capacity loss which cannot be compensated by the outer code. This is a fundamental result for the design of capacity achieving codes.

Chapter 3

Rate Matching by Rate-Compatible Convolutional Codes

Rate-Compatible Convolutional (RCC) (Turbo) codes are part of many physical layer specifications of modern mobile communication systems as, e.g., in the LTE specification [3GP11]. The code rate of a fixed-rate convolutional mother (Turbo) code is commonly adapted by simple bit puncturing or bit repetition. Bit puncturing allows to flexibly increase the effective code rate by discarding a certain number of encoded bits, while bit repetition facilitates lower effective code rates by duplicating a distinct number of encoded bits prior to transmission.

The combination of a fixed-rate convolutional code with bit puncturing is commonly referred to as *Rate-Compatible Punctured Convolutional* (RCPC) codes [CCG79, Hag88]. In the last decades, a lot of research has been done on RCPC codes to identify the optimal puncturing pattern for a specific coding system [HSS90, MK99, RM00, BGV05]. The fundamentals of these codes are briefly described in Section 3.1. In contrast to that, lower code rates are commonly provided by bit repetition referred to as *Rate-Compatible Repetition Convolutional* (RCRC) codes [KH90, LS00] (see Section 3.2).

An alternative class to the latter codes are *Rate-Compatible Insertion Convolutional* (RCIC) codes initially proposed by Collins et al. in [CH93] under the name *determinate state convolutional codes* (see Section 3.3). These codes construct lower code rates from a single fixed-rate mother code by inserting known bits, which are often called dummy or pilot bits, into the information bit sequence before convolutional encoding. These codes are also known as *pruned convolutional codes* [WC98] which is motivated by the effect of dummy bits on the trellis representation of these codes as illustrated in Section 3.3. Other terms, which are sometimes used in literature, are *dummy/pilot bit insertion*, *information nulling* or *code shortening*.

Following this concept, Xu and Romme presented in [XR00] a class of multi-rate convolutional codes which have surpassed conventional repetition schemes. Later on, the effectiveness of dummy bit insertion has also been demonstrated in the context of Turbo codes [KMFS03, KH08]; however, no comparison to competing rate matching schemes is given by the authors. In recent years, dummy bit insertion

has been adopted to many standardized communication systems. In [LPRP99] it has been proposed for the Korean mobile communication system, in [XM01] for speech transmission over GSM EDGE, in [BEV11] for the UMTS LTE system, and in [SM12] for the DVB family of standards.

While most of the contributions propose to insert dummy bits equidistantly into the information bit stream, Kadhim et al. [KH08] and Öberg et al. [OS97] suggest to insert these bits according to the distance spectrum of the employed mother code. This optimization provides performance improvements in particular for small dummy bit fractions.

As already mentioned, bit puncturing and bit repetition is employed in LTE in order to flexibly adapt the code rate to the instantaneous channel quality. In Section 3.4, the realization of channel coding and modulation in LTE is described which serves as the benchmark system throughout this thesis.

3.1 Rate-Compatible Punctured Convolutional Codes

Rate-Compatible Punctured Convolutional (RCPC) codes have firstly been proposed by Cain et al. in [CCG79] for code rates of $(i-1)/i$, $i \in \mathbb{N}$, and generalized by Hagenauer in [Hag88] as a procedure to flexibly adapt the code rate of a fixed-rate mother code to any rates higher than the code rate of the mother code. Due to its benefits like high flexibility and low complexity, it is part of many mobile systems as, e.g., LTE and HSPA, which are standardized by the *3rd Generation Partnership Project* (3GPP).

RCPC Encoder: The encoder structure is shown in Figure 3.1. The input bits $\mathbf{b} = (b_1, \ldots, b_m, \ldots, b_{\ell_b})$ are encoded by a convolutional encoder with generator polynomial(s) $\mathbf{G}_{\mathrm{CC}}$ and code rate r_{CC} resulting in the output bit stream $\mathbf{y} = (y_1, \ldots, y_n, \ldots, y_{\ell_y})$ of length $\ell_y = \ell_b \cdot r_{\mathrm{CC}}^{-1}$. Then, bit puncturing of rate $0 < r_{\mathrm{P}} < 1$ is employed yielding the vector $\breve{\mathbf{y}} = (\breve{y}_1, \ldots, \breve{y}_{\breve{n}}, \ldots, \breve{y}_{\ell_{\breve{y}}})$ of length $\ell_{\breve{y}} = \ell_y \cdot r_{\mathrm{P}}$. The code rate of the RCPC code is then given by

$$r_{\mathrm{PC}} = \frac{r_{\mathrm{CC}}}{r_{\mathrm{P}}} > r_{\mathrm{CC}}. \tag{3.1}$$

RCPC Decoder: The decoder structure is shown in Figure 3.2. Before *Soft-In/Soft-Out* (SISO) decoding, all channel-related *Log-Likelihood Ratio*s (LLRs) $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\dot{\mathbf{y}})$ that correspond to the punctured (non-transmitted) code bits $\dot{\mathbf{y}}$ are set to

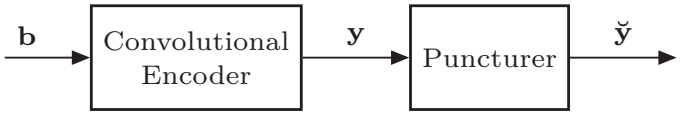


Figure 3.1: Structure of a RCPC encoder.

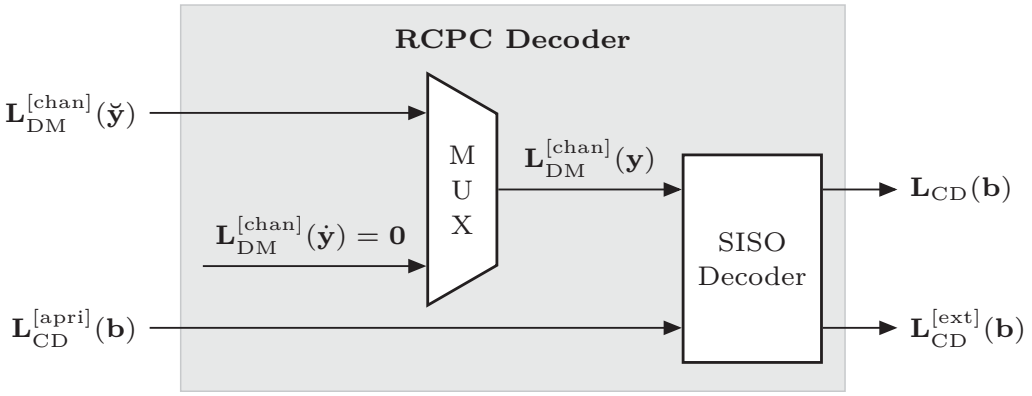


Figure 3.2: Structure of a RCPC decoder.

zero. This reduces the channel-related information provided to the SISO decoder. However, for adequate channel conditions, the remaining information regarding the code bits $\check{\mathbf{y}}$ is sufficient to retrieve all information bits. Therefore, puncturing is often employed in good channel conditions resulting in a reduced bandwidth consumption. If a Turbo coding system is considered, also *a priori* information on the information bits $\mathbf{b}$ is provided to the SISO decoder. This information is delivered by the constituent SISO decoder as described in Section 2.2.1.

Such a structure enables the utilization of one encoder/decoder pair for any puncturing rates r_{P} which is one major advantage of rate-compatible convolutional codes compared to rate matching schemes based on an ensemble of different codes with individual code rates.

3.2 Rate-Compatible Repetition Convolutional Codes

Like RCPC codes, *Rate-Compatible Repetition Convolutional* (RCRC) codes [Kal90, LS00] are also part of many standardized mobile communication systems. The code rate adaptation is carried out by duplicating a certain number of code bits after convolutional coding in order to provide additional channel-related information to the corresponding SISO decoder located at the receiver. There are many possible realizations of RCRC codes which differ in the utilized repeater. Conventionally, deterministic repetition schemes [3GP05, 3GP11, LS00] which repeat the output bits according to a predefined pattern are employed. However, random repeaters are also possible and are helpful for the information theoretic evaluation of RCRC codes.

RCRC Encoder: The structure of the corresponding encoder is shown in Figure 3.3. The input bits $\mathbf{b} = (b_1, \ldots, b_m, \ldots, b_{\ell_b})$ are encoded by a convolutional encoder with generator polynomial(s) $\mathbf{G}_{\mathrm{CC}}$ and code rate r_{CC} resulting in the output bit stream $\mathbf{y} = (y_1, \ldots, y_n, \ldots, y_{\ell_y})$ of length $\ell_y = \ell_b \cdot r_{\mathrm{CC}}^{-1}$. Then, bit repetition of rate $0 < r_{\mathrm{R}} \leq 1$ is employed yielding the vector $\check{\mathbf{y}} = (\check{y}_1, \ldots, \check{y}_{\check{n}}, \ldots, \check{y}_{\ell_{\check{y}}})$

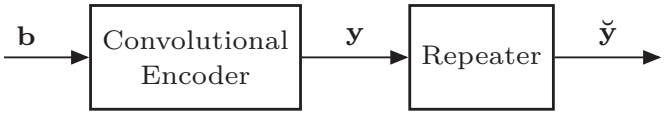


Figure 3.3: Structure of a RCRC encoder.

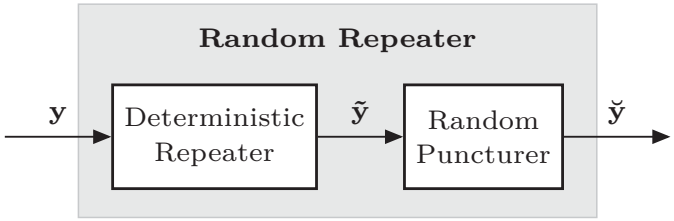


Figure 3.4: Realization of a random repeater.

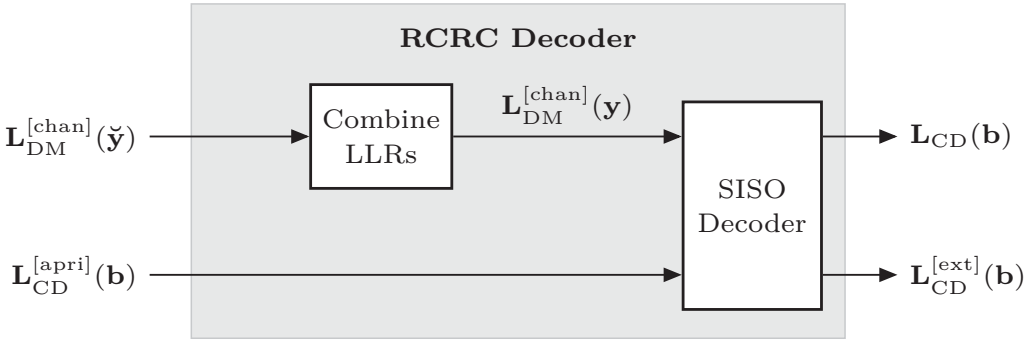


Figure 3.5: Structure of a RCRC decoder.

of length $\ell_{\breve{y}} = \ell_y \cdot r_{\mathrm{R}}^{-1}$. In case of a *random* repeater, this can be modeled by a serially concatenation of two components as shown in Figure 3.4. At first, a *deterministic* repeater with code rate

$$r_{\mathrm{DR}} = \frac{1}{Q} \quad , \quad Q \in \mathbb{N} \tag{3.2}$$

is employed which repeats all output bits of the convolutional encoder $Q-1$ times resulting in the vector $\tilde{\mathbf{y}} = (\tilde{y}_1, \ldots, \tilde{y}_{\tilde{n}}, \ldots, \tilde{y}_{\ell_{\tilde{y}}})$. Then, a puncturer *randomly* eliminates bits within $\tilde{\mathbf{y}}$ according to the puncturing rate $0 \leq r_{\mathrm{RP}} \leq 1$. Consequently, the puncturer introduces the random nature of the repetition scheme. Note that the puncturer only eliminates bits within the $Q-1$ repetitions of the frame $\mathbf{y}$, which guarantees that each bit is transmitted at least once.

The code rate of the RCRC code with random repetition is, thus, given as

$$r_{\mathrm{RC}} = r_{\mathrm{CC}} \cdot r_{\mathrm{R}} = \frac{\ell_b}{\ell_y} \cdot \frac{r_{\mathrm{DR}}}{r_{\mathrm{RP}}}, \quad \frac{r_{\mathrm{DR}}}{r_{\mathrm{RP}}} \leq 1 \tag{3.3}$$

taken into account the code rate of the convolutional code $r_{\mathrm{CC}} = \ell_b/\ell_y$ and the repetition rate of the random repeater $r_{\mathrm{R}} = r_{\mathrm{DR}}/r_{\mathrm{RP}}$.

RCRC Decoder: The decoder structure is shown in Figure 3.5. Before SISO decoding, all channel-related LLRs that correspond to the same code bit y_n of the convolutional encoder are accumulated (combined). This increases the amount of information on each code bit and results in a more reliable decoding of the information bits. If Turbo coding is considered, *a priori* information on the information bits $\mathbf{b}$ is exploited again.

3.3 Rate-Compatible Insertion Convolutional Codes

The concept of *Rate-Compatible Insertion Convolutional* (RCIC) codes has initially been introduced by Collins in [CH93], who has called these codes *determinate state* convolutional codes instead of RCIC codes. In [XR00], Xu and Romme addressed this concept again and developed a class of multirate codes which they called *insertion convolutional codes*. This class can simply construct lower rate codes based on a fixed-rate convolutional code (mother code) by inserting known bits (dummy bits) into the information bit sequence before convolutional encoding. In modern communication systems based on Turbo codes, in which two component codes are either serially or parallel concatenated, these codes are well suited for designing a rate-compatible, iterative coding system.

RCIC Encoder: The structure of an RCIC encoder is depicted in Figure 3.6. The multiplexed vector $\tilde{\mathbf{x}} = [\mathbf{b}, \mathbf{d}]$, containing the information bits $\mathbf{b}$ as well as the dummy bits $\mathbf{d} = (d_1, \ldots, d_l, \ldots, d_{\ell_d})$ $(1 \leq l \leq \ell_d)$, is either deterministically or even randomly interleaved by a bit interleaver π. However, most application scenarios, in which perfect information is exploited by a SISO decoder, have in common that deterministic interleavers are utilized, e.g., in [XR00, Ziv08, BLV+10, BV10b, BEV11, BCH09b]. If *Equal Error Protection* (EEP) is carried out, the most widely used deterministic interleaver distributes all known bits (dummy bits) ℓ_d equidistantly over the complete data block of length $\ell_x = \ell_b + \ell_d$. The realization of such an interleaver differs slightly between the applications known from literature, particularly for the case where the number of dummy bits is not an integer factor of the frame length. The interleaver proposed in this contribution can be realized

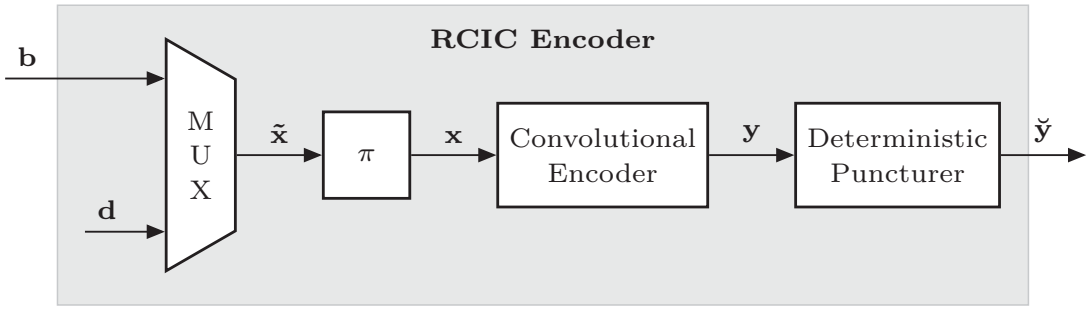


Figure 3.6: Structure of a RCIC encoder.

by means of Algorithm 3.1. The operator round $(\cdot)$ rounds its argument to the nearest integer value. This algorithm ensures the desired equidistant spacing of dummy bits within the interleaved data block $\mathbf{x}$ for the case $(\ell_x/\ell_d) \in \mathbb{N}$ and a at least "homogeneous" distribution for all other cases. A homogeneous distribution is achieved, e.g., in the following scenario: $\ell_d = 4$ dummy bits $\mathbf{d} = (d_1, \ldots, d_4) = (0, 0, 0, 0)$ are inserted into a block of $\ell_b = 6$ information bits $\mathbf{b} = (b_1, \ldots, b_6)$. Following the design rule of the proposed interleaver (Algorithm 3.1), the output block is given by

$$\mathbf{x} = (b_1, 0, b_2, b_3, 0, b_4, 0, b_5, b_6, 0). \tag{3.4}$$

The resulting sequence $\mathbf{x} = (x_1, \ldots, x_k, \ldots, x_{\ell_x})$ of length $\ell_x = \ell_b + \ell_d$ is then encoded by a convolutional encoder with code rate r_{CC} obtaining the output vector $\mathbf{y} = (y_1, \ldots, y_n, \ldots, y_{\ell_y})$ of length $\ell_y = \ell_x \cdot r_{\mathrm{CC}}^{-1}$. If the RCIC code is systematic, puncturing is performed to eliminate all systematic dummy bits. These bits are known in advance and do not have to be transmitted over the channel. Accordingly, the length of the output vector $\breve{\mathbf{y}} = (\breve{y}_1, \ldots, \breve{y}_{\breve{n}}, \ldots, \breve{y}_{\ell_{\breve{y}}})$ is reduced by the number of dummy bits ℓ_d resulting in $\ell_{\breve{y}} = \ell_x \cdot r_{\mathrm{CC}}^{-1} - \ell_d$. For non-systematic convolutional codes, no puncturing is employed, i.e., $\breve{\mathbf{y}} = \mathbf{y}$ and $\ell_{\breve{y}} = \ell_y$.

Hence, the code rate of an RCIC code is given by

$$r_{\mathrm{IC}} = \begin{cases} \frac{\ell_b}{\ell_b + \ell_d} \cdot r_{\mathrm{CC}} & : \quad \text{non-systematic RCIC code} \\ \frac{\ell_b}{(\ell_b + \ell_d) \cdot r_{\mathrm{CC}}^{-1} - \ell_d} & : \quad \text{systematic RCIC code} \end{cases} . \tag{3.5}$$

Due to $r_{\mathrm{IC}} \leq r_{\mathrm{CC}}$, RCIC codes are an alternative to RCRC codes and can be used instead for the design of a rate-compatible communication system.

RCIC Decoder: The structure of an RCIC decoder is depicted in Figure 3.7. The RCIC decoder receives the channel-related LLR vector $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\breve{\mathbf{y}})$ of the (punctured) output bits $\breve{\mathbf{y}}$. In the case that Turbo decoding is considered, *extrinsic* information on the input bits $\mathbf{b}$ is also provided to the RCIC decoder by the concatenated component decoder and exploited as *a priori* information $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}]}(\mathbf{b})$. The RCIC decoder inserts information on the dummy bits $\mathbf{d}$ to both input LLR vectors. While no channel-related information, i.e., $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{d}) = \mathbf{0}$, is given due to the

Algorithm 3.1 Dummy bit insertion

Initialize: $x_k = 0$ for $1 \leq k \leq \ell_x$
  for all m such that $1 \leq m \leq \ell_b$ **do**
    $x_k = b_m$ with $k = \mathrm{round}\left((m-1) \cdot \frac{\ell_b + \ell_d}{\ell_b}\right) + 1$
  end for
  (dummy bits $d_l = 0$ at all other position provided by initialization step)

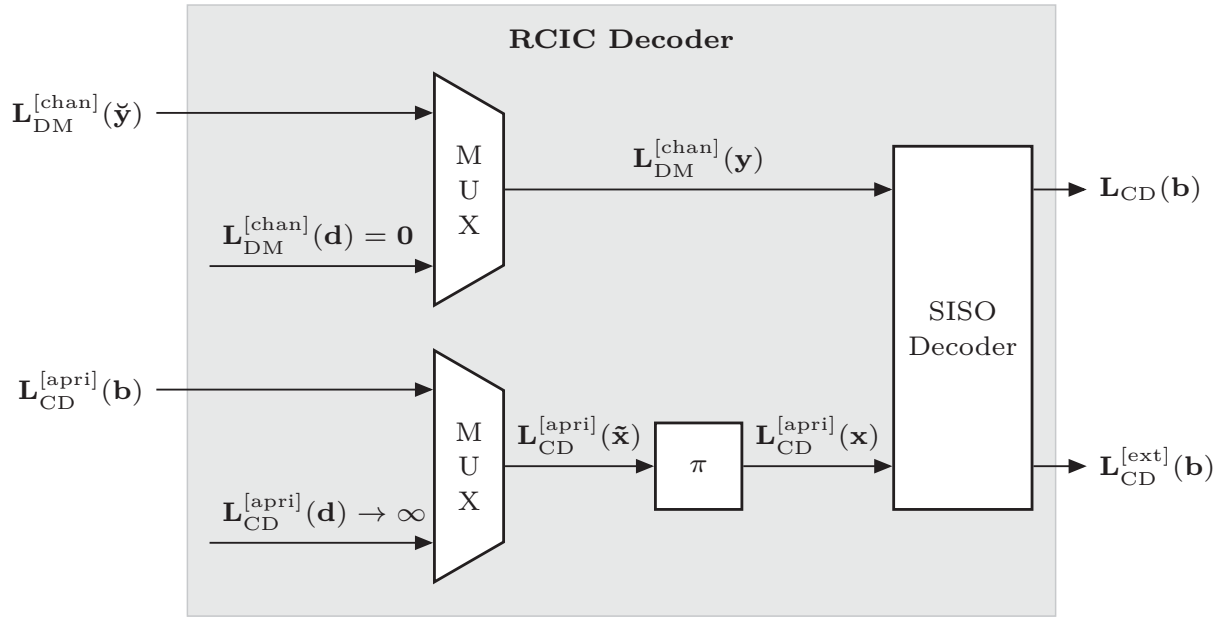
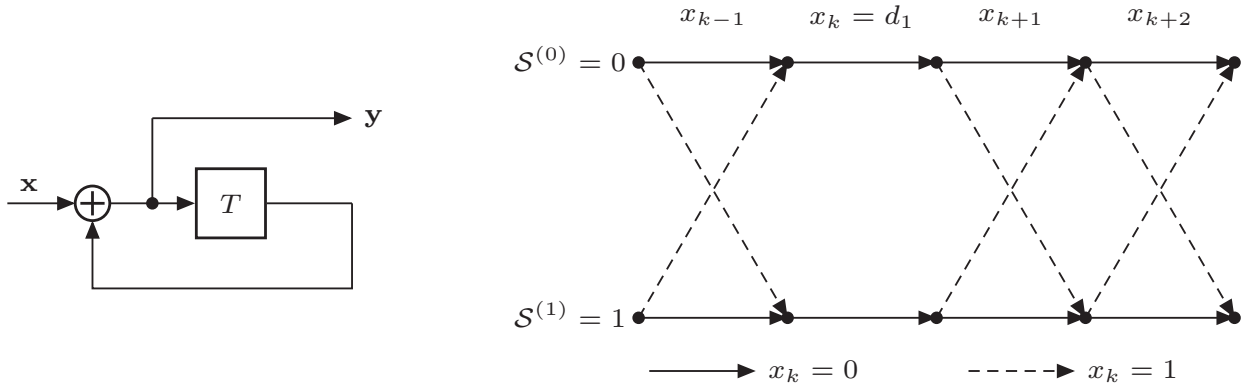


Figure 3.7: Structure of a RCIC decoder.

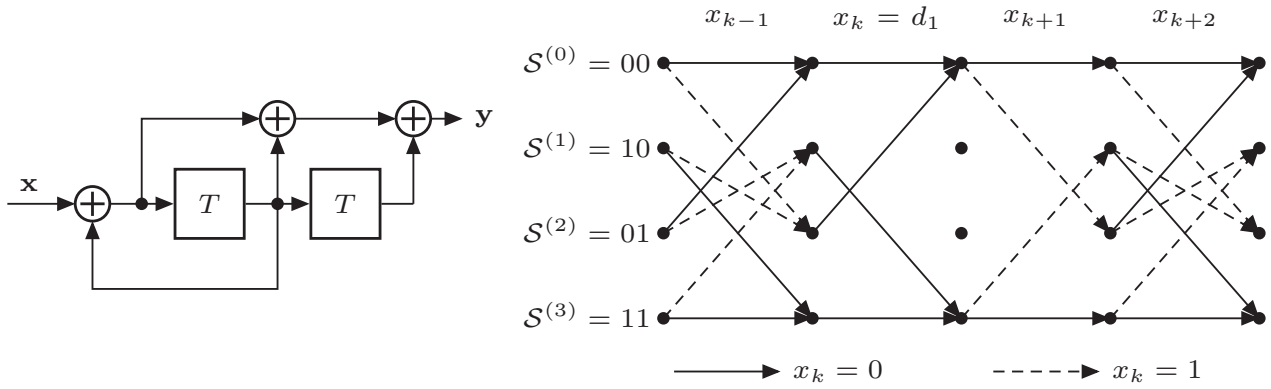
puncturing at the encoder side, perfect *a priori* information $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}]}(\mathbf{d}) \to \infty$ can be provided to the SISO decoder about the dummy bits since they are known in advance. The dummy bits can then be exploited as supporting points during the *Bahl-Cocke-Jelinek-Raviv* (BCJR) [BCJR74] decoding of the input bits $\mathbf{b}$ as it is visualized by the trellis diagrams in Figures 3.8 and 3.9.

As an example, one dummy bit d_1 is inserted at time instance k resulting in the input sequence $\mathbf{x} = (\ldots, x_{k-1}, d_1, x_{k+1}, x_{k+2}, \ldots)$ which is encoded by a rate-1 *Recursive Non-Systematic Code* (RNSC) with octal generator polynomial $\mathbf{G_1} = (2/3)_8$ and constraint length $\mathfrak{L}_1 + 1 = 2$ as well as a rate-1 RNSC with octal generator polynomial $\mathbf{G_2} = (7/6)_8$ and constraint length $\mathfrak{L}_2 + 1 = 3$. Due to the knowledge of the dummy bit, some transitions in the trellis diagrams can be discarded, i.e., the trellis diagram is *pruned* at each dummy bit position which motivates the term rate-compatible *pruned* convolutional codes. Besides the information provided by the pruned trellis, channel-related information on the parity bits at each dummy bit time instance can be exploited by the SISO decoder. Both together results in a more reliable estimation of the surrounding information bits and, thus, in a significantly improved decoding performance. The influence of the insertion of known bits on the *extrinsic* information has, e.g., been measured in [Cle06]. It has been observed that the influence diminishes for distances larger than approximately 2 times the constraint length of the code. From this consideration, it can be concluded that an equidistant distribution of dummy bits will be an appropriate choice for EEP, since the information provided by the dummy bits is homogeneously spread over the complete frame. For *Unequal Error Protection* (UEP), the dummy bits can be placed besides those bits which are most sensitive with respect to its impact on the signal quality after decoding.



(a) Block diagram. **(b)** Trellis diagram.

Figure 3.8: Type 1 convolutional code with octal generator polynomial $\mathbf{G_1} = (2/3)_8$ and constraint length $\mathfrak{L}_1 + 1 = 2$.



(a) Block diagram. **(b)** Trellis diagram.

Figure 3.9: Type 2 convolutional code with octal generator polynomial $\mathbf{G_1} = (7/6)_8$ and constraint length $\mathfrak{L}_1 + 1 = 3$.

For a general view on path pruning, the reader is referred to [WC98, WC02, WL08, BFC00, HvD06, HvDH+07, HHvD+10]. Based on a fixed-rate mother code, sub-codes with different code rates are systematically constructed by means of path pruning to allow for UEP. Path-pruning by means of dummy bit insertion is only one possible realization.

Decoding Complexity: Besides the performance gain, there is a complexity reduction achieved due to the omitted state transitions in the trellis diagram as we have shown in [BV10b]. Considering Figure 3.8 at first, the dummy bit $d_1 = 0$ effects a reduction of the complexity at time instance k by more than a factor of 2, since half of the state transitions can be discarded (pruned trellis) and no computation of the *extrinsic* information has to be carried out. The overall complexity reduction can even be increased for RCIC codes with merging trellis paths

Operation	information bit	dummy bit	
		type 1	type 2
$\max^*$	$4 \cdot 2^{\mathfrak{L}} - 2$	$2 \cdot 2^{\mathfrak{L}}$	0
ADD	$8 \cdot 2^{\mathfrak{L}} + 5$	$4 \cdot 2^{\mathfrak{L}} + 2$	2

Table 3.1: Complexity per information bit and dummy bit for the LogMAP channel decoder ($\mathfrak{L}$: length of the shift register).

at dummy bit positions (Figure 3.9). In this case, half of the final states cannot be reached due to the merged paths. Consequently, the number of state transitions at the succeeding time instance $k+1$ can be cut in half which reduces significantly the complexity at this time instance. It has been shown in [XR00] that merging paths at dummy bit positions can only be realized for recursive convolutional codes in which the last memory position is not connected to the feedback loop (type 2 codes, cf., Figure 3.9). Otherwise, all states can be reached by a dummy bit yielding a full trellis at time instance $k+1$ (type 1 codes, cf., Figure 3.8). The complexity introduced by an information bit and a dummy bit, respectively, is given in Tab. 3.1 and a detailed derivation is provided in Appendix C.

It has been distinguished between two types of operations, namely the ADD-operation and the $\max^*$-operation. The $\max^*$-operation is a fundamental operator in the BCJR algorithm (see Appendix C) and comprises a max-operation, 3 additions as well as 2 table lookups. In general, the computational time required for each operation depends on the processor architecture. In the following it is assumed that the complexity of one $\max^*$-operation is six times higher than the complexity of the ADD-operation. The increase in complexity depends on the constraint length $\mathfrak{L}+1$, i.e., on the code structure, as well as on the inserted fraction of dummy bits. According to Tab. 3.1, the relative complexity increase introduced by the dummy bit insertion $\Delta\mathcal{C} = \mathcal{C}^{[\mathrm{IC}]}/\mathcal{C}^{[\mathrm{CC}]} - 1$ is exemplary computed for the convolutional codes depicted in Figures 3.8 and 3.9 (index corresponds to code type) and for dummy bit fractions of $\ell_d/\ell_x = (1/3,\ 1/2,\ 2/3)$:

$$\frac{\ell_d}{\ell_x} = \frac{1}{3}: \quad \Delta\mathcal{C}_1 = \frac{\frac{1}{3}\ell_x \cdot (6 \cdot 4 + 10) + \frac{2}{3}\ell_x \cdot (6 \cdot 6 + 21)}{\frac{2}{3}\ell_x \cdot (6 \cdot 6 + 21)} - 1 \approx 29.8\,\%,$$

$$\Delta\mathcal{C}_2 \approx 0\,\%,$$

$$\frac{\ell_d}{\ell_x} = \frac{1}{2}: \quad \Delta\mathcal{C}_1 = \frac{\frac{1}{2}\ell_x \cdot (6 \cdot 4 + 10) + \frac{1}{2}\ell_x \cdot (6 \cdot 6 + 21)}{\frac{1}{2}\ell_x \cdot (6 \cdot 6 + 21)} - 1 \approx 59.6\,\%,$$

$$\Delta\mathcal{C}_2 \approx 0\,\%,$$

$$\frac{\ell_d}{\ell_x} = \frac{2}{3}: \quad \Delta\mathcal{C}_1 = \frac{\frac{2}{3}\ell_x \cdot (6 \cdot 4 + 10) + \frac{1}{3}\ell_x \cdot (6 \cdot 6 + 21)}{\frac{1}{3}\ell_x \cdot (6 \cdot 6 + 21)} - 1 \approx 119.3\,\%,$$
$$\Delta\mathcal{C}_2 \approx 0\,\%.$$

Accordingly, a considerable increase in the overall computational complexity is caused by the insertion of dummy bits if employing convolutional codes of type 1. However, this can be avoided by the utilization of type 2 codes providing a low complexity rate matching scheme [BV10b].

3.4 Adaptive Modulation and Coding in LTE

LTE release 10 and beyond is evaluated by several partners of the 3GPP as candidate for the *International Mobile Telecommunications - Advanced* (IMT-Advanced) standard of the *International Telecommunication Union* (ITU). Therefore, its physical layer will act as a reference for many practical and theoretic evaluations concerning the physical layer aspects that are investigated in this theses.

In general, rate matching in the LTE standard is realized by bit puncturing and bit repetition after convolutional Turbo coding according to Figure 3.10. At first, 24 *Cyclic Redundancy Check* (CRC) bits are appended to a frame of $\ell_{\bar{b}}$ information bits $\bar{\mathbf{b}}$ which is then encoded by a systematic rate-1/3 Turbo encoder consisting of two *Parallel Concatenated Convolutional Code*s (PCCCs) with octal generator

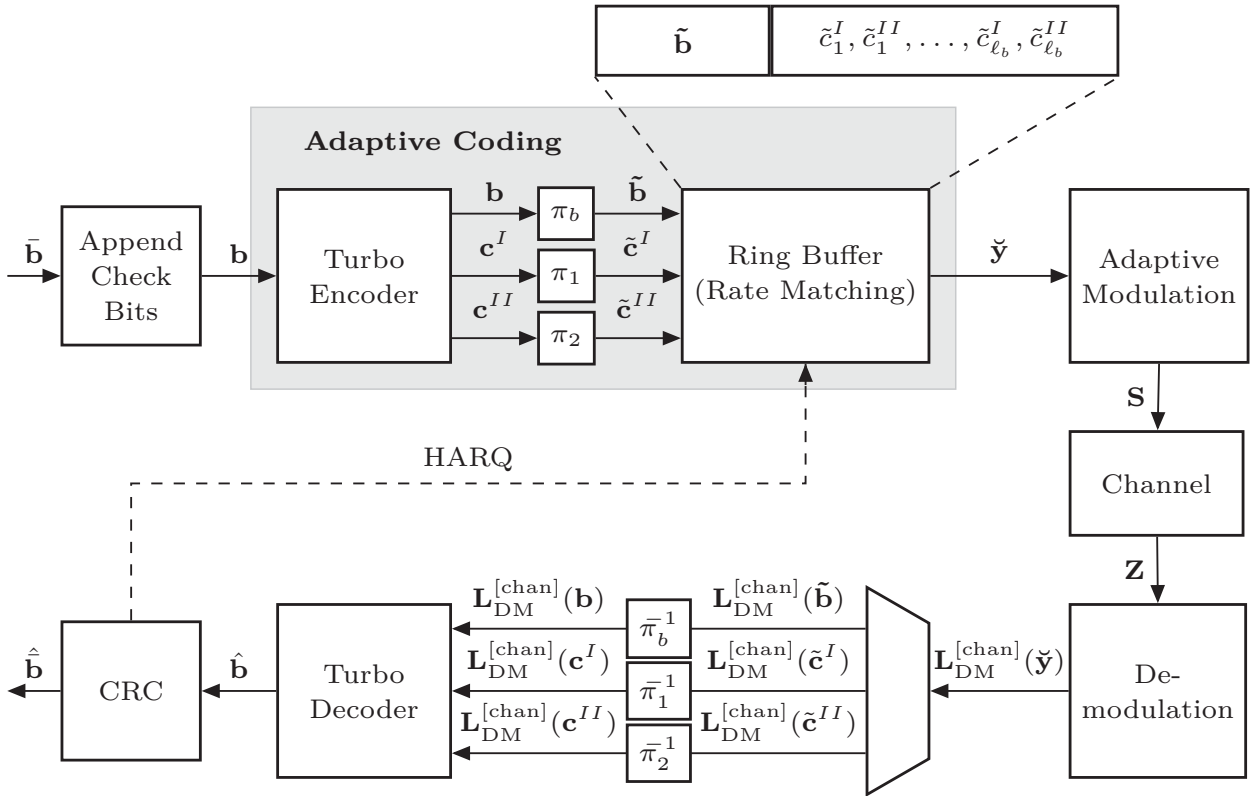


Figure 3.10: Model of the LTE physical layer.

polynomials $\mathbf{G}_{\mathrm{CC}} = (1, 15/13)_8$ and constraint length $\mathfrak{L} + 1 = 4$, each generating one parity bit per information bit.

The encoded bits are then separated into three streams: The first contains the systematic, i.e., the uncoded information bits $\mathbf{b} = (b_1, \ldots, b_m, \ldots, b_{\ell_b})$, while the second and third contain the parity bits of the two constituent encoders $\mathbf{c}^I = (c_1^I, \ldots, c_m^I, \ldots, c_{\ell_b}^I)$ and $\mathbf{c}^{II} = (c_1^{II}, \ldots, c_m^{II}, \ldots, c_{\ell_b}^{II})$, respectively. For an efficient and easy to implement rate matching, all streams are individually interleaved by so-called sub-interleavers π_b, π_1 and π_2 and written to a ring buffer. At first, all systematic bits $\tilde{\mathbf{b}}$ are written to the ring buffer. Then the parity bits of both streams $\tilde{\mathbf{c}}^I$ and $\tilde{\mathbf{c}}^{II}$ are interlaced and also written to the ring buffer according to the structure shown in Figure 3.10. The ring buffer is visualized as a vector which is continuously read out starting at the beginning again if the end of the buffer is reached. Finally, a block $\breve{\mathbf{y}} = (\breve{y}_1, \ldots, \breve{y}_{\breve{n}}, \ldots, \breve{y}_{\ell_{\breve{y}}})$ of $\ell_{\breve{y}}$ encoded bits is selected for transmission resulting in an effective code rate $r_{\mathrm{RM}} = \ell_b / \ell_{\breve{y}}$ after rate matching. The size $\ell_{\breve{y}}$ of the block of encoded bits is determined by a scheduler according to the user's instantaneous channel quality, the user's requested throughput, maximum delay, target BER and the current load of the radio cell. Thereby, the scheduler implicitly influences the code rate r_{RM} of the user. A block size $\ell_{\breve{y}} < 3 \cdot \ell_b$ results in a code rate $r_{\mathrm{RM}} > 1/3$, whereas if $\ell_{\breve{y}}$ is sufficiently large, the code rate r_{RM} can take values $r_{\mathrm{RM}} < 1/3$ by repetition of systematic and parity bits. Consequently, bit puncturing and bit repetition is implicitly performed by reading out the ring buffer. For details of the rate matching algorithm the reader is referred to [3GP11].

The bits selected for transmission are grouped to vectors of $I \in \{2, 4, 6\}$ bits, which are assigned to complex modulation symbols $\mathbf{S}$ according to the specified modulation schemes QPSK, 16QAM, or 64QAM. After adaptive modulation, the complex modulation symbols $\mathbf{S}$ are transmitted over the channel using *Orthogonal Frequency-Devision Multiplexing* (OFDM).

On the receiving side, OFDM demodulation is performed. The demodulated complex symbols $\mathbf{Z}$ are fed into a soft demapper which delivers reliability information in form of LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\tilde{\mathbf{b}})$, $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\tilde{\mathbf{c}}^I)$, and $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\tilde{\mathbf{c}}^{II})$ on the interleaved, systematic information bits $\tilde{\mathbf{b}}$ and the parity bits of the two constituent encoders $\tilde{\mathbf{c}}^I$, $\tilde{\mathbf{c}}^{II}$, respectively. The deinterleaved LLRs are then passed on to the Turbo decoder consisting of two parallel concatenated SISO channel decoders using the symbol-by-symbol *Maximum A Posteriori* estimation in the *Logarithmic domain* (LogMAP) algorithm [BCJR74, RVH95] for soft channel decoding. After a fixed number of decoding iterations, $\hat{\mathbf{b}}$ is estimated from the resulting *a posteriori* LLRs[1].

If the CRC detects an erroneous frame, the receiver requests for a transmission of additional (incremental) redundancy about the same frame by sending a *Negative*

[1] It is worth mentioning that the receiver is not part of the standardization. There are only minimum performance requirements which have to be fulfilled. Nevertheless, the above described receiver realization is assumed throughout this theses.

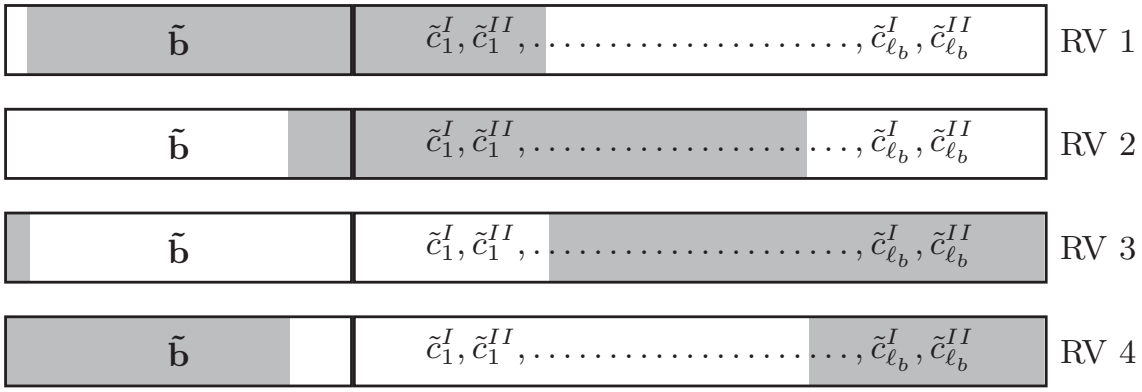


Figure 3.11: Redundancy versions defined in LTE for the example of $\ell_b = 6144$ and $r_{\mathrm{RM}} = 2/3$ ($\ell_{\breve{y}} = 9216$).

ACKnowledge (NACK) to the transmitter. Otherwise, an *ACKnowledge* (ACK) is fed back resulting in the transmission of consecutive (new) frames. The feedback channel is only indicated by the dashed line in Figure 3.10. In many contributions to HARQ, an ideal feedback channel is assumed, however, in real scenarios the system has to cope with an unreliable feedback channel. The influence of such an non-ideal feedback channel on the system performance will be analyzed in Chapter 6. The LTE *Hybrid Automatic Repeat reQuest* (HARQ) scheme allows for up to $K = 4$ transmissions of different combinations of systematic and parity bits, the so-called *Redundancy Versions* (RVs). Obviously, each RV transmission implicitly decreases the effective code rate and results in losses of throughput and latency. According to [3GP11], the initial reading position θ_ρ in the ring buffer of a distinct RV ρ ($1 \leq \rho \leq K$) is given by

$$\theta_\rho = \Psi\left(2(\rho-1)\left\lceil\frac{3(\ell_b+4)}{8\Psi}\right\rceil+2\right)+1,\ \Psi = \left\lceil\frac{\ell_b+4}{32}\right\rceil, \tag{3.6}$$

where $\lceil\cdot\rceil$ rounds up its argument to the nearest integer value. Two important conclusions can be drawn from (3.6):

(a) The initial reading position of the first RV is given by $\theta_1 = 2 \cdot \Psi + 1 \neq 1$, i.e., the first RV does not start at the beginning of the ring buffer.

(b) The ring buffer is not read out continuously, i.e., the current RV ρ does not start at the end of the previous RV $\rho-1$. This may result in a rate matching procedure performing bit puncturing and bit repetition at the same time due to overlapping RVs.

Both conclusions are visualized in Figure 3.11 for the example of $\ell_b = 6144$ and $r_{\mathrm{RM}} = 2/3$, i.e., $\ell_{\breve{y}} = 9216$ bits are read out from the ring buffer. Accordingly, the initial RV starts at position $\breve{n} = \theta_1 = 386$ and ends at position $\breve{n} = 9602$ which is larger than the initial reading position of the second RV $\breve{n} = \theta_2 = 5018$.

Chapter 4

EXIT Functions of Rate-Compatible Codes

Rate-compatible codes are employed in many transmission systems in order to cope with bit errors caused by typical impairments on the physical link (fading, shadowing, doppler spread, multipath propagation, etc.). Either rate-compatible *Low-Density Parity-Check* (LDPC) codes or rate-compatible Turbo codes consisting of two either parallel or serially concatenated convolutional codes are commonly used for physical layer channel coding in state of the art systems [IEE09,IEE04,3GP11]. Rate matching in LDPC based systems is conventionally performed by means of a set of LDPC codes. Each LDPC code is designed for a specific target code rate which is supported by the transmission system. The desired code rate adaptation is, thus, achieved by changing the employed LDPC code during transmission. In contrast to that, rate matching in Turbo coded systems is commonly carried out by bit puncturing (higher code rate) or bit repetition (lower code rate) based on a fixed-rate mother code (see Chapter 3). An alternative to bit repetition is dummy bit insertion, which is a potential candidate for future standards in wireless communication.

In the major part of this chapter, the effectiveness of the above mentioned realizations of rate-compatible Turbo codes are evaluated in an information theoretic context. Usually, the influence of these schemes on the system performance is analyzed by means of extensive Monte-Carlo simulations. Slight modifications of the employed schemes necessitate new simulations in order to obtain a performance evaluation of these modifications. In contrast to that, a new analysis is proposed to predict the expected system behavior dependent on the fraction of additional or punctured bits.

In particular the theoretic evaluation of *Rate-Compatible Insertion Convolutional* (RCIC) codes is of great benefit. Besides the application of rate matching, RCIC codes and their underlying principles, i.e., the insertion of known bits (dummy bits) into the information bit sequence before convolutional encoding, attract more and more attention in recent studies related to other research fields. In [BV10b], we have exploited dummy bits in a system based on *Iterative Source-Channel Decoding* (ISCD) [AVS01, Gör00] in order to decrease the overall complexity under the constraint of an equal overall code rate and a com-

parable system performance. The underlying concept is further applied to joint channel coding and cryptography, e.g., in [RZ08, Ziv08, ZRR09a, ZRR09b] where iterative decryption is performed based on two information blocks which are individually encrypted but jointly encoded by the channel encoder. If one block can be decrypted error-free, this information can be exploited as perfect *a priori* information at the *Soft-In/Soft-Out* (SISO) decoder during the iterative decryption process of the second block. In [ZT08], one information block is even substituted by dummy bits which are known in advance and can, thus, already be exploited during the first decryption attempt of the second block. In channel estimation, pilot symbols are mostly used for estimation of the current channel state, since a blind estimation imposes a high complexity and suffers from a performance degradation. However, instead of inserting pilots at the modulation stage, in pilot symbol assisted coding schemes, a predetermined fraction of dummy bits (pilot bits) is inserted into the information bit sequence before channel encoding [MS01,KHG+05,KGHL05,OKPL08,KH08,BCH09b,BCH09a,MK09]. If a systematic code is employed, these bits, which are sometimes called *encoded pilots*, appear unchanged at the output of the channel encoder and can directly be used for channel estimation. Since the channel encoder produces additionally a certain number of parity bits for each pilot bit, pilot symbol assisted coding schemes provide a joint rate matching and channel estimation functionality. A further potential application of the principle of dummy bit insertion must be seen in the area of packet-switched transmission. Packet headers, which are mainly responsible for an error-free packet routing, are attached to the payload. These headers are kept as small as possible in order to limit the loss in spectral efficiency. In [BLV+10], a packet-switched multimedia transmission system based on cross-layer communication has been considered, which can further reduce the loss in spectral efficiency. After error-free header decoding, all header bits are known and perfect reliability information can be fed back to the SISO channel decoder. Consequently, the header bits take on the role of dummy bits within the Turbo decoding resulting in an improved decoding of the multimedia content. This header bit assisted decoding principle will be addressed in Chapter 7 in more detail. All these applications have in common that perfect information about a fraction of bits is exploited during iterative channel decoding to improve the decoding of the remaining bits. Therefore, all these schemes will highly benefit from an information theoretic analysis.

A widely used information theoretic tool for the prediction of the convergence behavior of Turbo codes is the analysis by *EXtrinsic Information Transfer* (EXIT) charts [tB00b,tB01a]. As shown in Section 2.2.3, EXIT charts illustrate the flow of *extrinsic* information through the constituent SISO decoders. If the rate matching scheme can be accurately modeled by random processes, it is possible to derive analytical expressions which approximate the influence of the respective rate matching scheme on the EXIT chart of the underlying mother code. Such an information

theoretic evaluation has been presented in [Tho07] for *Rate-Compatible Punctured Convolutional* (RCPC) (Turbo) codes with random puncturing and pre-published in [BV11a] and [BV11b] for *Rate-Compatible Repetition Convolutional* (RCRC) and RCIC (Turbo) codes with random repetition and random insertion, respectively. This EXIT chart analysis provides a prediction of the expected gains in terms of *extrinsic* mutual information based on the EXIT charts of the underlying mother code and can be used for a semi-analytically optimized rate matching procedure. The basic derivations of all expressions are revisited in Section 4.1, Section 4.2, and Section 4.3, respectively. Furthermore, it is shown that even the performance of the deterministic rate matching scheme employed in the LTE system can be approximated by a model based on random processes with sufficient accuracy. Finally, an information theoretic analysis also supports the comparison of competing realizations of rate-compatible Turbo codes as, e.g., RCRC and RCIC Turbo codes with respect to their convergence behavior (Section 4.4). This significantly reduces the number of required EXIT chart simulations. In Section 4.5, it has been illustrated that this analysis is also applicable to LDPC codes employing rate matching by means of bit puncturing, bit repetition, or dummy bit insertion. Examples are shown for both code classes under the assumption that transmission is carried out over a *Binary Erasure Channel* (BEC) or a *Binary-Input Additive White Gaussian Noise Channel* (BIAWGNC). However, the presented EXIT chart analysis can also simply be adapted to the *Binary Symmetric Channel* (BSC), which is demonstrated in Section 4.5 as well.

4.1 Rate-Compatible Punctured Convolutional Codes

In [Tho07], an information theoretical analysis of RCPC codes with random puncturing has been presented based on the transmission model introduced by Ashikhmin et al. in [AKtB02]. More precisely, the EXIT function of a systematic RCPC code with mother code $\mathbf{G}_{\text{sys}} = (1, \mathbf{G}_{\text{nsys}})$ is determined based on the EXIT function of a non-systematic convolutional component code with generator polynomial $\mathbf{G}_{\text{nsys}}$. Furthermore, the analysis has been presented for a transmission over a BEC. In Section 4.1.1, at first, this analysis will be recalled briefly for a slightly simplified task in which the EXIT function of an RCPC code is derived by means of the employed convolutional mother code. Note that this is only a special case of the analysis presented in [Tho07]. However, it will be extended to the case that transmission is carried out over a BIAWGNC. Based on these results, a novel semi-analytical optimization of this rate matching scheme subject to a given target code rate is presented in Section 4.1.2. Using the more flexible transmission model proposed in [Tho07], conventional deterministic puncturing schemes can be modeled more accurately as exemplarily shown for the puncturing scheme employed in the LTE standard (see Section 4.1.3).

4.1.1 Random Puncturing

In the following, it has to be distinguished between component codes of *Parallel Concatenated Convolutional Code*s (PCCCs) and *Serially Concatenated Convolutional Code*s (SCCCs).

Parallel Concatenation

The information theoretic transmission model that incorporates the puncturing effect is illustrated in Figure 4.1. It is based on the model presented in Section 2.2.3 which is depicted in Figure 4.2 again for comparison.

According to Figure 4.1, the information bits $\mathbf{b}$ are encoded by the convolutional component encoder yielding the output bits $\mathbf{y}$. After puncturing with rate r_P, the remaining (unpunctured) bits $\breve{\mathbf{y}}$ are transmitted over the communication channel with mutual information $\mathcal{I}_\mathrm{C}$.

For RCPC codes, random puncturing with rate r_P ($0 \leq r_\mathrm{P} \leq 1$) can be modeled by a BEC with capacity $C_\mathrm{P} = r_\mathrm{P}$ as illustrated in Figure 4.1. If no puncturing is employed ($r_\mathrm{P} = 1$), the capacity of the BEC is identical to one. In the other extreme case, i.e., all bits are punctured ($r_\mathrm{P} = 0$), the capacity of the BEC is equal to zero. In between, the BEC ensures that no information is given for the fraction of punctured bits, while perfect information is given for the rest of the bits $\breve{\mathbf{y}}$ which are selected for transmission.

The serial concatenation of the BEC with capacity $C_\mathrm{P} = r_\mathrm{P}$ and the communication channel with mutual information $\mathcal{I}_\mathrm{C}$ can be interpreted as an *equivalent* communication channel with mutual information

$$\mathring{\mathcal{I}}_\mathrm{C} = r_\mathrm{P} \cdot \mathcal{I}_\mathrm{C}. \tag{4.1}$$

Compared to the communication channel in the information theoretic transmission model of the convolutional mother code (Figure 4.2), the *equivalent* communication channel in Figure 4.1 delivers a modified vector $\mathbf{L}_\mathrm{DM}^{[\mathrm{chan}],\circ}(\mathbf{y})$ of channel-related LLRs to the SISO decoder. In this vector, a fraction of r_P LLRs are identical to zero, while a fraction of $(1-r_\mathrm{P})$ LLRs follow the probability distribution associated with

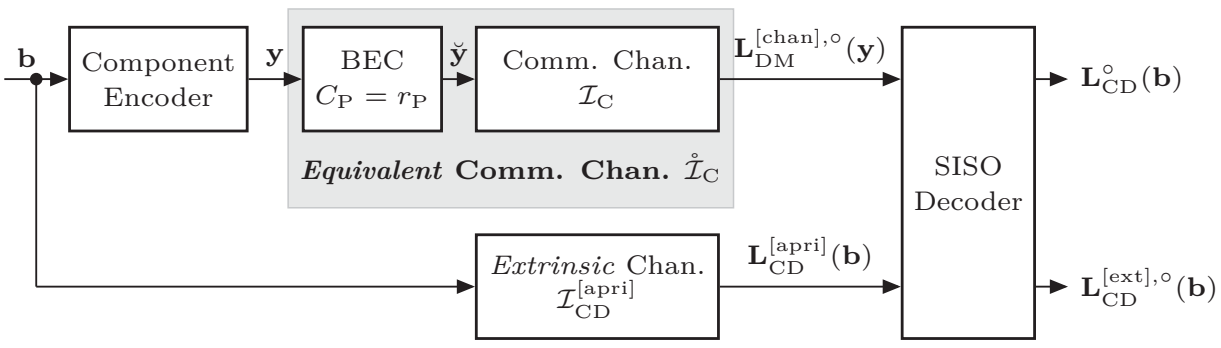


Figure 4.1: Information theoretic transmission model for RCPC codes.

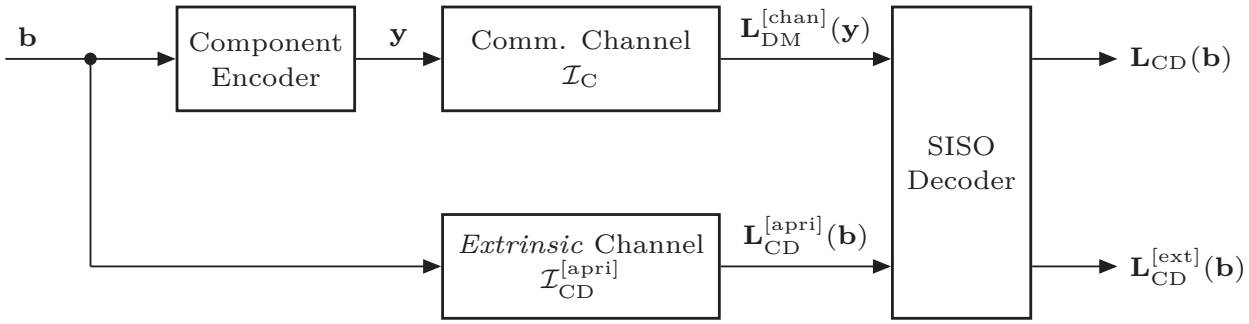


Figure 4.2: Information theoretic transmission model for the convolutional mother code.

the considered communication channel model. Furthermore, the SISO decoder receives the *a priori* LLRs $\mathbf{L}_{CD}^{[apri]}(\mathbf{b})$, which are delivered by the *extrinsic* channel, and generates modified *extrinsic* LLRs $\mathbf{L}_{CD}^{[ext],\circ}(\mathbf{b})$ and *a posteriori* LLRs $\mathbf{L}_{CD}^{\circ}(\mathbf{b})$. It is worth mentioning that the SISO decoder is identical for a system with and without puncturing, since it is fully determined by the employed convolutional component code.

The aim of this section is to determine the EXIT function of an RCPC code based on the EXIT function of the underlying convolutional mother code. To this end, it is assumed that the *equivalent* communication channel can be modeled by the same channel model as the communication channel, yielding output LLRs which follow the same probability distribution. Only the mutual information of both channels is assumed to be different according to (4.1). It is evident that this assumption is only valid for a communication channel which can be modeled as a BEC; a serial concatenation of two BECs can be substituted by one BEC with adapted capacity. In contrast to that, e.g., a serial concatenation of a BEC with a communication channel modeled as a BIAWGNC will not provide Gaussian distributed output LLRs.

However, for the BEC case, the EXIT function $\mathcal{T}_{CD}^{(PC)}$ of the RCPC code with random puncturing can be constructed by means of the EXIT function $\mathcal{T}_{CD}^{(CC)}$ of the underlying convolutional mother code by simply shifting the channel-related mutual information from $\mathcal{I}_C$ to $\mathring{\mathcal{I}}_C$:

$$\mathcal{T}_{CD}^{(PC)}\left(\mathcal{I}_{CD}^{[apri]}\middle|\mathcal{I}_C\right) = \mathcal{T}_{CD}^{(CC)}\left(\mathcal{I}_{CD}^{[apri]}\middle|\mathring{\mathcal{I}}_C\right), \tag{4.2}$$

with $0 \leq \mathcal{I}_{CD}^{[apri]} \leq 1$, $0 \leq \mathcal{I}_C \leq 1$, $0 \leq \mathring{\mathcal{I}}_C \leq 1$ and $\mathring{\mathcal{I}}_C$ given according to (4.1).

Although (4.2) is only valid for a transmission over a BEC, it will be shown that it also provides a close approximation for the BIAWGNC case.

For the BEC and the BIAWGNC, EXIT charts are usually parameterized by the erasure probability P_C and the channel SNR E_s/N_0 of the communication channel, respectively.

In the BEC case, the mutual information of the communication channel and

the *equivalent* communication channel, respectively, are given by

$$\mathcal{I}_{\mathrm{C}} = 1 - P_{\mathrm{C}}, \tag{4.3}$$

$$\mathring{\mathcal{I}}_{\mathrm{C}} = 1 - \mathring{P}_{\mathrm{C}}, \tag{4.4}$$

where $\mathring{P}_{\mathrm{C}}$ denotes the erasure probability of the *equivalent* communication channel. Taking further (4.1) into account, the following relation is valid for the BEC case:

$$1 - \mathring{P}_{\mathrm{C}} = \mathring{\mathcal{I}}_{\mathrm{C}} = r_{\mathrm{P}} \cdot \mathcal{I}_{\mathrm{C}} = r_{\mathrm{P}} \cdot (1 - P_{\mathrm{C}}). \tag{4.5}$$

Solving (4.5) for $\mathring{P}_{\mathrm{C}}$ results in

$$\mathring{P}_{\mathrm{C}} = 1 - r_{\mathrm{P}} \cdot (1 - P_{\mathrm{C}}). \tag{4.6}$$

Hence, it is possible to determine the EXIT function of the RCPC code by simulating the EXIT function of the convolutional mother code for a transmission over a BEC with increased erasure probability $\mathring{P}_{\mathrm{C}} \geq P_{\mathrm{C}}$.

Accordingly, the EXIT function of the RCPC code for a transmission over a BIAWGNC with variance $\sigma_n^2 = N_0/2$ ($N_0/2$ denotes the spectral noise power density) can be approximated by the EXIT function of the convolutional mother code simulated at an increased spectral noise power density $\mathring{\sigma}_n^2 = \mathring{N}_0/2$. By means of (2.33) and (2.34), the mutual information of the communication channel and the *equivalent* communication channel can be expressed as

$$\mathcal{I}_{\mathrm{C}} = J\left(\frac{2}{\sigma_n}\right) = J\left(2 \cdot \sqrt{2 \cdot \frac{E_s}{N_0}}\right), \tag{4.7}$$

$$\mathring{\mathcal{I}}_{\mathrm{C}} = J\left(\frac{2}{\mathring{\sigma}_n}\right) = J\left(2 \cdot \sqrt{2 \cdot \frac{E_s}{\mathring{N}_0}}\right). \tag{4.8}$$

By means of (4.1), a relation connecting the SNR $E_s/\mathring{N}_0$ of the *equivalent* communication channel with the SNR E_s/N_0 of the communication channel can be derived according to[1]

$$\begin{aligned} &\Leftrightarrow J\left(2 \cdot \sqrt{2 \cdot \frac{E_s}{\mathring{N}_0}}\right) = r_{\mathrm{P}} \cdot J\left(2 \cdot \sqrt{2 \cdot \frac{E_s}{N_0}}\right) \\ &\Leftrightarrow \frac{E_s}{\mathring{N}_0} = \frac{1}{8} \cdot \left[J^{-1}\left(r_{\mathrm{P}} \cdot J\left(2 \cdot \sqrt{2 \cdot \frac{E_s}{N_0}}\right)\right)\right]^2. \end{aligned} \tag{4.9}$$

The Equations (4.6) and (4.9) are fundamental results for the semi-analytical rate matching procedure proposed in Section 4.1.2. Both equations will be adapted to the respective rate matching scheme throughout this chapter.

[1] For AWGN transmission with QPSK, E_s/N_0 and $E_s/\mathring{N}_0$ have to be substituted by $0.5 \cdot E_s/N_0$ and $0.5 \cdot E_s/\mathring{N}_0$, respectively. For higher modulation schemes, the relation between $\mathcal{I}_{\mathrm{C}}$ and E_s/N_0 has to be measured.

Example

The systematic rate-1/2 component code of the LTE Turbo code (parallel concatenation, see Section 3.4) with octal generator polynomials $\mathbf{G}_{\mathrm{CC}} = (1, 15/13)_8$ and constraint length $\mathfrak{L} + 1 = 4$ is considered for the following set up:

(a) Communication channel: BEC with erasure probability $P_{\mathrm{C}} = 0.4$, *extrinsic* channel: BEC,

(b) communication channel: BIAWGNC with $E_s/N_0 = -1.5\,\mathrm{dB}$, *extrinsic* channel: BIAWGNC.

The corresponding EXIT functions are depicted in Figure 4.3 for different target code rates $r_{\mathrm{PC}} \in \{2/3, 3/4, 4/5\}$. They are derived (bold lines) based on the EXIT function of the underlying mother code according to (4.2) and compared with the EXIT functions measured by Monte-Carlo simulation (dashed lines).

Considering Figure 4.3(a) (BEC case), the derived EXIT functions match perfectly their measured versions for all three target code rates r_{PC}. As stated above, minor deviations can be observed in the BIAWGNC case for all considered target code rates due to the deviation of the actual probability distribution of the LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}],\circ}(\mathbf{y})$ from the assumed Gaussian distribution. For a decreased puncturing rate r_{P}, the influence of the BEC is increased (see Figure 4.1) resulting in higher deviations between the probability distributions and, thus, in slightly increased deviations between the simulated and derived EXIT functions.

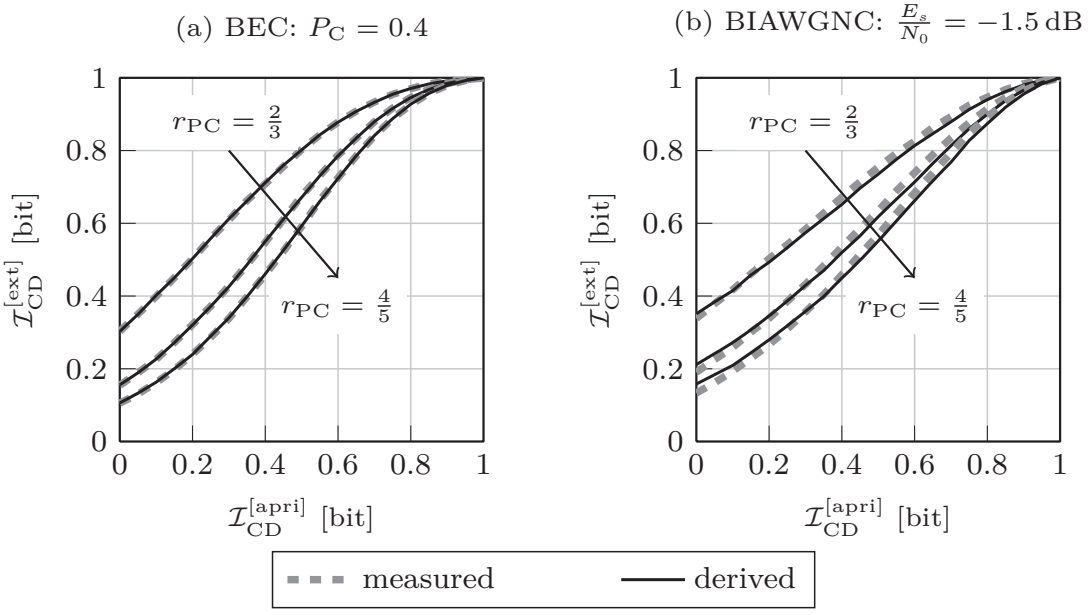


Figure 4.3: Comparison of derived and measured EXIT functions for the LTE component code and different target code rates $r_{\mathrm{PC}} \in \{2/3, 3/4, 4/5\}$.

Serial Concatenation

In the case of serial concatenated convolutional codes, it has to be distinguished between inner and outer codes. The desired EXIT function $\mathcal{T}_{\text{in}}^{(\text{PC})}$ of an inner RCPC code can directly be determined based on the EXIT function $\mathcal{T}_{\text{in}}^{(\text{CC})}$ of the underlying inner convolutional mother code using the results of the previous section:

$$\mathcal{T}_{\text{in}}^{(\text{PC})}\left(\mathcal{I}_{\text{in}}^{[\text{apri}]}\middle|\,\mathcal{I}_{\text{C}}\right)=\mathcal{T}_{\text{in}}^{(\text{CC})}\left(\mathcal{I}_{\text{in}}^{[\text{apri}]}\middle|\,\mathring{\mathcal{I}}_{\text{C}}\right), \tag{4.10}$$

with $0\leq\mathcal{I}_{\text{in}}^{[\text{apri}]}\leq 1$, $0\leq\mathcal{I}_{\text{C}}\leq 1$, and $0\leq\mathring{\mathcal{I}}_{\text{C}}\leq 1$.

For RCPC codes employed as outer codes, the *extrinsic* channel in Figure 2.9 is substituted by an *equivalent extrinsic* channel with mutual information

$$\mathcal{I}_{\text{out}}^{[\text{apri}],\circ}=r_{\text{P}}\cdot\mathcal{I}_{\text{out}}^{[\text{apri}]} \tag{4.11}$$

which consists of a BEC with capacity $C_{\text{P}}=r_{\text{P}}$ modeling the influence of the random puncturing and the serially concatenated *extrinsic* channel with mutual information $\mathcal{I}_{\text{out}}^{[\text{apri}]}$.

The EXIT function $\mathcal{T}_{\text{out}}^{(\text{PC})}$ of an outer RCPC code employed in a serially concatenated coding scheme can then be determined by means of the EXIT function $\mathcal{T}_{\text{out}}^{(\text{CC})}$ of the convolutional mother code according to

$$\mathcal{T}_{\text{out}}^{(\text{PC})}\left(\mathcal{I}_{\text{out}}^{[\text{apri}]}\right)=\mathcal{T}_{\text{out}}^{(\text{CC})}\left(\mathcal{I}_{\text{out}}^{[\text{apri}],\circ}\right) \tag{4.12}$$

with $0\leq\mathcal{I}_{\text{out}}^{[\text{apri}]}\leq 1$, $0\leq\mathcal{I}_{\text{out}}^{[\text{apri}],\circ}\leq 1$ and $\mathcal{I}_{\text{out}}^{[\text{apri}],\circ}$ given according to (4.11). It is worth noting again that equality in (4.10) as well as (4.12) is only given in the BEC case. For a transmission over a BIAWGNC, this is only an approximate solution.

It can be concluded from (4.11) that $\mathcal{I}_{\text{out}}^{[\text{apri}],\circ}<1$ for $\mathcal{I}_{\text{out}}^{[\text{apri}]}=1$ and $r_{\text{P}}<1$. Therefore, random puncturing ($r_{\text{P}}<1$) according to Figure 4.1 should not be employed after outer encoding in SCCC schemes since the loss in mutual information due to puncturing can not be compensated during decoding even if perfect *a priori* information on the remaining bits is provided to the decoder. In this case, the loss in mutual information is given by $1-r_{\text{P}}$ which means that the top right corner cannot be reached by the EXIT chart. A more detailed discussion of random puncturing for outer convolutional codes is carried out in [Tho07].

4.1.2 Optimization by EXIT Charts

EXIT charts are well suited to predict the behavior of Turbo codes within the waterfall region avoiding extensive, time-consuming *Bit Error Rate* (BER) simulations. In the case of rate matching, an EXIT chart analysis can help to optimize the system subject to its BER performance and spectral efficiency dependent on the

instantaneous channel condition. The aim is to determine the *minimum* puncturing rate $r_{\mathrm{P}}^{\min}$ which allows for near error-free decoding at a specific channel quality P_{C} or E_s/N_0, taking into account the number of Turbo iterations. So far, this task could only be solved by carrying out extensive Monte Carlo simulations for each target channel quality individually. A novel semi-analytical solution, which is less time-consuming, can now be formulated for random puncturing by means of the expressions derived in the previous sections. This procedure is summarized in Algorithm 4.1.

As an example, the parallel concatenated, systematic rate-1/3 convolutional

Algorithm 4.1 RCPC Turbo codes: Optimization by EXIT charts

Aim: *Determine the minimum puncturing rate r_P^{min} required for near error-free transmission at a given channel quality E_s/N_0 and with a given maximum number of Turbo iterations.*

(1) Determine, by means of EXIT-chart simulations, the channel quality at which error-free transmission becomes possible for the employed mother Turbo code, taking the number of Turbo iterations into account. This channel quality corresponds to the channel quality $\mathring{P}_{\mathrm{C}}$ or $E_s/\mathring{N}_0$ of the *equivalent* communication channel.

(2) Transform $\mathring{P}_{\mathrm{C}}$ or $E_s/\mathring{N}_0$ to the mutual information $\mathring{\mathcal{I}}_{\mathrm{C}}$ of the *equivalent* communication channel according to (4.4) or (4.8):

$$\mathring{\mathcal{I}}_{\mathrm{C}} = \begin{cases} 1 - \mathring{P}_{\mathrm{C}} & : \text{ BEC} \\ J\left(2 \cdot \sqrt{2 \cdot \frac{E_s}{\mathring{N}_0}}\right) & : \text{ BIAWGNC} \end{cases} .$$

(3) Compute the mutual information $\mathcal{I}_{\mathrm{C}}$ of the communication channel for the target channel qualities P_{C} or E_s/N_0 by means of (4.3) and (4.7), respectively:

$$\mathcal{I}_{\mathrm{C}} = \begin{cases} 1 - P_{\mathrm{C}} & : \text{ BEC} \\ J\left(2 \cdot \sqrt{2 \cdot \frac{E_s}{N_0}}\right) & : \text{ BIAWGNC} \end{cases} .$$

(4) Compute the puncturing rate $r_{\mathrm{P}}^{\min}$ which is required to obtain the mutual information $\mathring{\mathcal{I}}_{\mathrm{C}}$ of the *equivalent* communication channel using (4.1):

$$r_{\mathrm{P}}^{\min} = \frac{\mathring{\mathcal{I}}_{\mathrm{C}}}{\mathcal{I}_{\mathrm{C}}}.$$

Turbo code employed in the LTE standard is considered. Near error-free transmission shall be performed over a BIAWGNC at $E_s/N_0 = -2\,\mathrm{dB}$ with a maximum of 8 Turbo iterations. At first, Mote-Carlo simulations are carried out to determine the EXIT chart of the mother code which allows for near error-free transmission with approximately 8 Turbo iterations. The corresponding channel quality $E_s/\mathring{N}_0 = -4.6\,\mathrm{dB}$ amounts to the channel quality of the *equivalent* communication channel depicted in Figure 4.1. In a second step, the corresponding mutual information $\mathring{\mathcal{I}}_\mathrm{C}$ is computed by means of ten Brink's J-function according to $\mathring{\mathcal{I}}_\mathrm{C} = J\left(2 \cdot \sqrt{2 \cdot E_s/\mathring{N}_0}\right) \approx J(1.67) \approx 0.376$. Similarly, the mutual information of the communication channel is determined to $\mathcal{I}_\mathrm{C} = J\left(2 \cdot \sqrt{2 \cdot E_s/N_0}\right) \approx J(2.25) \approx 0.565$. Finally, performing step (4) results in a minimum puncturing rate of $r_\mathrm{P}^\mathrm{min} \approx 2/3$ and, thus, in an effective code rate after Turbo coding and rate matching of $r_\mathrm{PCCC} \approx 1/2$.

The resulting EXIT chart of the semi-analytically optimized RCPC Turbo code (bold lines) and the decoding trajectory (dashed line) are illustrated in Figure 4.4 (left hand side). As desired, approximately 8 Turbo iterations are sufficient for near error-free decoding which is also verified by the corresponding BER depicted in Figure 4.4 as well (right hand side). The bit error rate at the target channel quality $E_s/N_0 = -2\,\mathrm{dB}$ is marked by a circle. In this simulation, the frame size has been set to $\ell_b = 100000$ bits.

The semi-analytical optimization process and its potential becomes quite clear

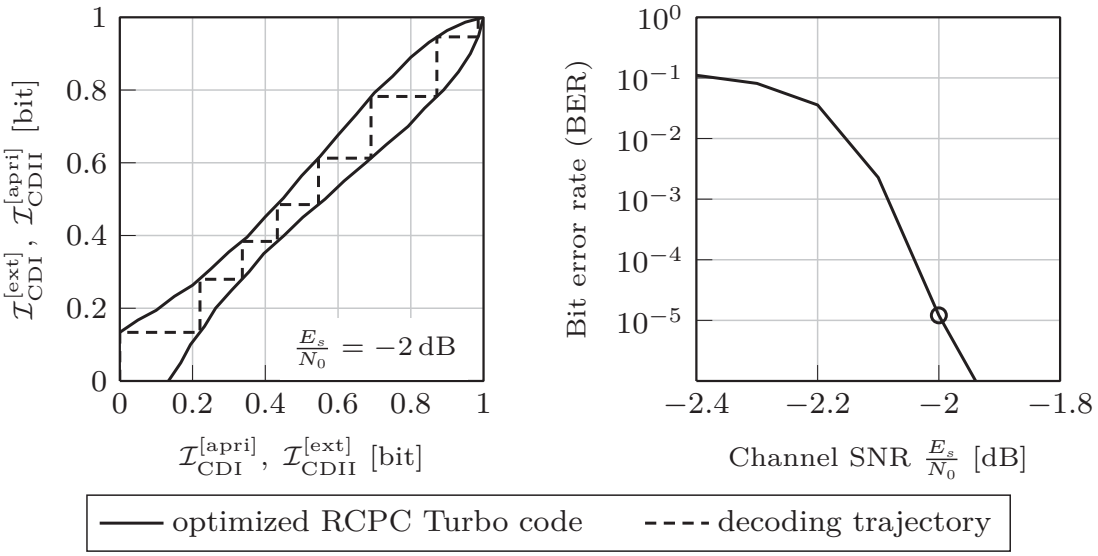


Figure 4.4: EXIT chart optimization at $E_s/N_0 = -2\,\mathrm{dB}$ for the systematic rate-1/3 LTE Turbo code. Left hand side: EXIT chart of the semi-analytically optimized RCPC Turbo code (bold lines) including its decoding trajectory (dashed line). Right hand side: corresponding BER.

by the following remarks:

- The EXIT chart after optimized puncturing at channel quality $E_s/N_0 = -2\,\mathrm{dB}$ can be approximated by the EXIT chart of the mother code at $E_s/\mathring{N}_0 = -4.6\,\mathrm{dB}$ with high accuracy.
- The EXIT chart of the mother code has to be determined only once in advance.
- EXIT chart optimized rate matching can then be performed on the fly without introducing considerable computational complexity, i.e., the influence of the instantaneous channel quality can directly be transformed to the optimal puncturing rate $r_{\mathrm{P}}^{\mathrm{min}}$.

4.1.3 Deterministic Puncturing

In standardized transmission systems, rate matching is commonly performed using deterministic puncturing schemes as, e.g., specified in the LTE standard. Although this cannot be modeled by means of information theory, transmission models based on random processes can provide close approximations. The deterministic rate matching procedure of LTE is described in Section 3.4. The output bits of the systematic rate-1/3 Turbo encoder are initially stored in a ring buffer according to Figure 3.11. The first redundancy version does not start at the beginning of the ring buffer, which results in the puncturing of a small fraction of systematic bits if the target code rate after rate matching is smaller than the code rate of the employed Turbo code. Consequently, the fraction of punctured parity bits $\mathbf{c}$ and punctured systematic bits $\mathbf{b}$ may differ. It is evident that the distinct puncturing pattern cannot be modeled by an information theoretic model, however, it is at least possible to account for the different fractions of punctured bits.

The transmission model of the previous section can be adapted according to Figure 4.5 in order to allow for an individual puncturing of both types of bits. This model complies with the EXIT chart model analyzed in [Tho07] and is highly suited to model the behavior of the LTE puncturing scheme more precisely than the model depicted in Figure 4.1.

The systematic bits at the output of the convolutional component encoder with octal generator polynomial $\mathbf{G}_{\mathrm{sys}} = (1, \mathbf{G}_{\mathrm{nsys}})$ and code rate r_{CC} are separated from the parity bits and individually transmitted over the communication channel after puncturing with rate r''_{P}. This provides additional *a priori* information on the information bits $\mathbf{b}$ to the SISO decoder. The parity bits generated by the residual non-systematic encoder with octal generator polynomial $\mathbf{G}_{\mathrm{nsys}}$ and code rate $r_{\mathrm{CC}}^{[\mathrm{nsys}]} = 1/(r_{\mathrm{CC}}^{-1} - 1)$ are punctured by a puncturer with a, in general, different puncturing rate r'_{P} prior to transmission. The LLR vectors $\mathbf{L}_{\mathrm{DM,mod}}^{[\mathrm{chan}],\circ}(\mathbf{c})$ and $\mathbf{L}_{\mathrm{CD,mod}}^{[\mathrm{apri}]}(\mathbf{b})$ at the output of the *equivalent* communication channel with mutual information $\mathring{\mathcal{I}}_{\mathrm{C,mod}}$ and the *equivalent extrinsic* channel with mutual information

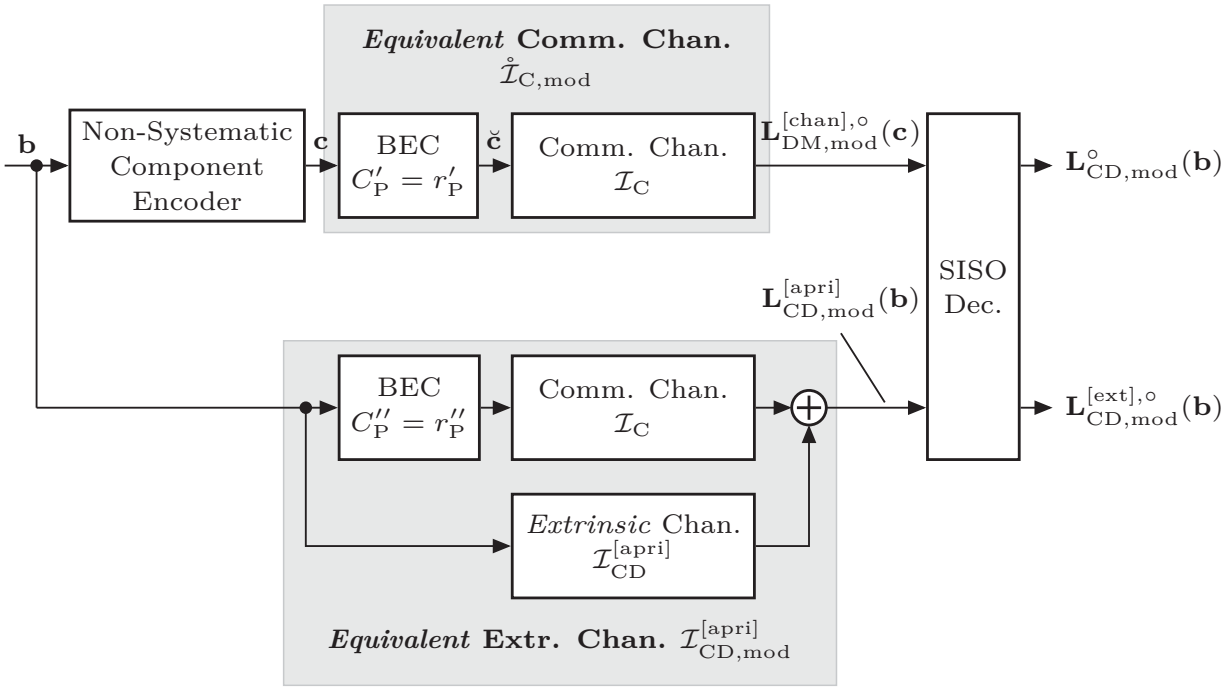


Figure 4.5: Modified information theoretic transmission model approximating the effect of the deterministic LTE bit puncturing scheme.

$\mathcal{I}^{[\mathrm{apri}]}_{\mathrm{CD,mod}}$, respectively, are finally exploited by the SISO decoder to compute *extrinsic* LLRs $\mathbf{L}^{[\mathrm{ext}],\circ}_{\mathrm{CD,mod}}(\mathbf{b})$ and *a posteriori* LLRs $\mathbf{L}^{\circ}_{\mathrm{CD,mod}}(\mathbf{b})$.

Modeling the communication channel and the *extrinsic* channel as a BEC, the EXIT function $\mathcal{T}^{(\mathrm{PC})}_{\mathrm{CD,\,mod}}$ of the RCPC code based on the modified transmission model can be derived from the EXIT function $\mathcal{T}^{(\mathrm{nsys})}_{\mathrm{CD}}$ of the non-systematic convolutional component code according to [Tho07, tB01b]:

$$\mathcal{T}^{(\mathrm{PC})}_{\mathrm{CD,\,mod}}\left(\mathcal{I}^{[\mathrm{apri}]}_{\mathrm{CD}}\middle|\,\mathcal{I}_{\mathrm{C}}\right) = \mathcal{T}^{(\mathrm{nsys})}_{\mathrm{CD}}\left(\mathcal{I}^{[\mathrm{apri}]}_{\mathrm{CD,mod}}\middle|\,\mathring{\mathcal{I}}_{\mathrm{C,mod}}\right), \tag{4.13}$$

where $\mathcal{I}^{[\mathrm{apri}]}_{\mathrm{CD,mod}}$ and $\mathring{\mathcal{I}}_{\mathrm{C,mod}}$ are given by

$$\mathcal{I}^{[\mathrm{apri}]}_{\mathrm{CD,mod}} = \begin{cases} \mathcal{I}^{[\mathrm{apri}]}_{\mathrm{CD}} + r''_{\mathrm{P}} \cdot \mathcal{I}_{\mathrm{C}} \cdot \left(1 - \mathcal{I}^{[\mathrm{apri}]}_{\mathrm{CD}}\right) & : \ \mathrm{BEC} \\ J\left(\sqrt{\left[J^{-1}\left(\mathcal{I}^{[\mathrm{apri}]}_{\mathrm{CD}}\right)\right]^2 + \left[J^{-1}\left(r''_{\mathrm{P}} \cdot \mathcal{I}_{\mathrm{C}}\right)\right]^2}\right) & : \ \mathrm{BIAWGNC} \end{cases} \tag{4.14}$$

$$\mathring{\mathcal{I}}_{\mathrm{C,mod}} = r'_{\mathrm{P}} \cdot \mathcal{I}_{\mathrm{C}}. \tag{4.15}$$

A close approximation is again given for a communication channel modeled as BIAWGNC, although the probability distributions at the output of the *equivalent* communication channel and the *equivalent extrinsic* channel does not follow the assumed Gaussian distribution.

It is worth noting that the transmission model shown in Figure 4.1 is included in the modified transmission model as the special case $r'_{\mathrm{P}} = r''_{\mathrm{P}}$. Furthermore, (4.13), (4.14), and (4.15) provide a construction rule for the EXIT function of a systematic convolutional code if the EXIT function of the corresponding non-systematic code is given (special case $r'_{\mathrm{P}} = r''_{\mathrm{P}} = 1$). Consequently, the modified transmission model enables a more general view on the behavior of randomly punctured convolutional codes. Based on the modified transmission model, the target code rate r_{PCCC} after PCCC coding and puncturing is given in dependence of r'_{P} and r''_{P} according to

$$r_{\mathrm{PCCC}} = \frac{1}{r''_{\mathrm{P}} + 2 \cdot r_{\mathrm{CC}}^{[\mathrm{nsys}]} \cdot r'_{\mathrm{P}}}. \tag{4.16}$$

The performance of the modified random puncturing as described above and the deterministic puncturing scheme employed in the LTE standard are compared in terms of their EXIT functions in Figure 4.6. The puncturing of the systematic bits and the parity bits depends on the initial reading position within the ring buffer and, thus, on the block size ℓ_b according to (3.6). For the maximum block size $\ell_b = 6144$ specified in the LTE standard, the ring buffer is read out starting at the initial reading position $\theta_1 = 386$. Consequently, a fraction of 6.28 % of the systematic bits is initially punctured. The considered target code rates after rate-1/3 convolutional Turbo coding and puncturing are chosen to $r_{\mathrm{PCCC}} \in \{1/2, 2/3\}$. Taking (4.16) into account, this results in the puncturing rates $r''_{\mathrm{P}} = 1 - \theta_1/\ell_b \in \{0.937, 0.937\}$ and $r'_{\mathrm{P}} \in \{0.5315, 0.2815\}$. Transmission is exemplarily performed over a BEC with erasure probabilities $P_{\mathrm{C}} = 0.4$ and over a

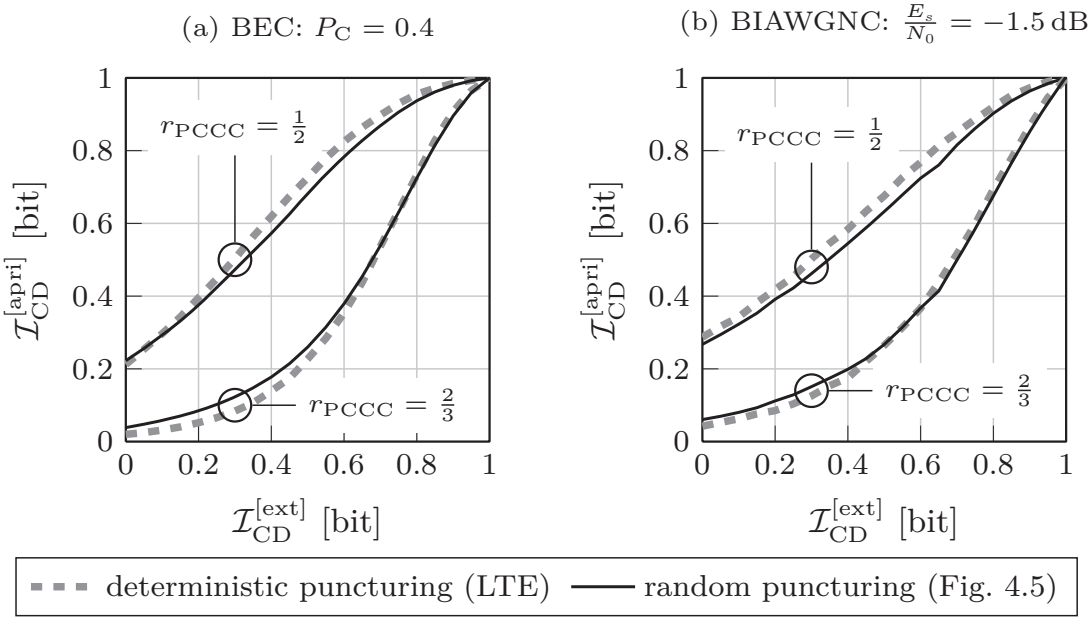


Figure 4.6: Random puncturing according to Figure 4.5 versus deterministic puncturing according to [3GP11] for the rate-1/2 LTE component code.

BIAWGNC with $E_s/N_0 = -1.5\,\mathrm{dB}$. As illustrated in Figure 4.6, small deviations are observable between the deterministic LTE puncturing scheme and the random puncturing according to the model proposed in [Tho07] and depicted in Figure 4.5. It signifies that a similar performance can be expected in both cases. Consequently, the theoretical results are beneficial for a performance prediction and optimization of the LTE system reducing significantly the number of Monte-Carlo simulations. However, this optimization requires more EXIT chart simulations due to the additional degree of freedom (individual puncturing of systematic bits and parity bits). One puncturing effect has to be evaluated by means of Monte-Carlo simulations and only the other puncturing effect can be analyzed using the expressions derived in this section. A joint optimization is impossible.

4.2 Rate-Compatible Repetition Convolutional Codes

Bit repetition is often employed in conventional communication systems in order to flexibly decrease the effective code rate after fixed-rate Turbo coding. Its influence on the overall system performance can be visualized using EXIT charts. In this section, the influence of bit repetition is modeled by means of information theory and analyzed similarly to the previous section. For random bit repetition, expressions for the EXIT charts after rate matching can be derived as shown in Section 4.2.1. These expressions are then used for a semi-analytical optimization of the considered rate matching scheme avoiding a considerable number of time-consuming simulations (see Section 4.2.2). Furthermore, in Section 4.2.3 it is sketched that the general model introduced in Section 4.2.1 can be adapted for a better prediction of the behavior of the deterministic bit repetition scheme specified in the LTE standard.

4.2.1 Random Repetition

For the random repeater described in Section 3.2 it is possible to derive its EXIT function by means of the EXIT function of the employed convolutional mother code as we have shown in [BV11a]. The basic approach is identical to that one described in the previous section for random puncturing. At first, an *equivalent* communication channel will be defined which includes both, the communication channel and the effect of the bit repetition. Then, the mutual information of that channel will be derived analytically. Finally, the EXIT function of the RCRC code will be determined based on the assumption that the output of the communication channel and the output of the *equivalent* communication channel follow the same probability distribution. The above sketched derivation has again been carried out for *Parallel Concatenated Convolutional Codes* (PCCCs) and *Serially Concatenated Convolutional Codes* (SCCCs).

Parallel Concatenation

The random bit repetition can be incorporated into the information theoretic transmission model in Figure 4.2 by substituting the communication channel with mutual information $\mathcal{I}_{\mathrm{C}}$ by an *equivalent* communication channel with increased mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ as shown in Figure 4.7. The corresponding model of the *equivalent* communication channel is given in Figure 4.8. The information bits $\mathbf{b}$ are encoded by the convolutional component encoder resulting in the bit stream $\mathbf{y}$. As described in Section 3.2, the random repeater can be modeled by a serial concatenation of a deterministic repeater of repetition rate $r_{\mathrm{DR}} = 1/Q$ ($Q \in \mathbb{N}$) and a random puncturer of puncturing rate $0 \leq r_{\mathrm{RP}} \leq 1$. Accordingly, the deterministic repetition can be interpreted as a transmission over $Q = r_{\mathrm{DR}}^{-1}$ independent memoryless communication channels with mutual information $\mathcal{I}_{\mathrm{C}}$, while the effect of the random puncturer is introduced by $Q-1$ generally different BECs with capacity

$$C_{\mathrm{RP}}^{(q)} = r_{\mathrm{RP}}^{(q)} \quad \text{and} \quad 2 \leq q \leq Q, \quad C_{\mathrm{RP}}^{(1)} = r_{\mathrm{RP}}^{(1)} = 1, \tag{4.17}$$

where $r_{\mathrm{RP}}^{(q)}$ denotes the puncturing rate in branch q. Note that $r_{\mathrm{RP}}^{(1)} = 1$ since only the repeated fraction of bits are punctured, i.e., $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}],1}(\mathbf{y}) = \mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{y})$. The overall puncturing rate is then given as

$$r_{\mathrm{RP}} = r_{\mathrm{DR}} \cdot \sum_{q=1}^{Q} r_{\mathrm{RP}}^{(q)} = \frac{1}{Q} \sum_{q=1}^{Q} r_{\mathrm{RP}}^{(q)}. \tag{4.18}$$

Summation of all Q partial channel-related *Log-Likelihood Ratio* (LLR) vectors $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}],q}(\mathbf{y})$ yields the overall LLR vector $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}],\Sigma}(\mathbf{y})$ which is exploited by the SISO decoder:

$$\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}],\Sigma}(\mathbf{y}) = \sum_{q=1}^{Q} \mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}],q}(\mathbf{y}). \tag{4.19}$$

Obviously, the repetition code generates an increased amount of mutual information between the code bits $\mathbf{y}$ and their corresponding LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}],\Sigma}(\mathbf{y})$ at

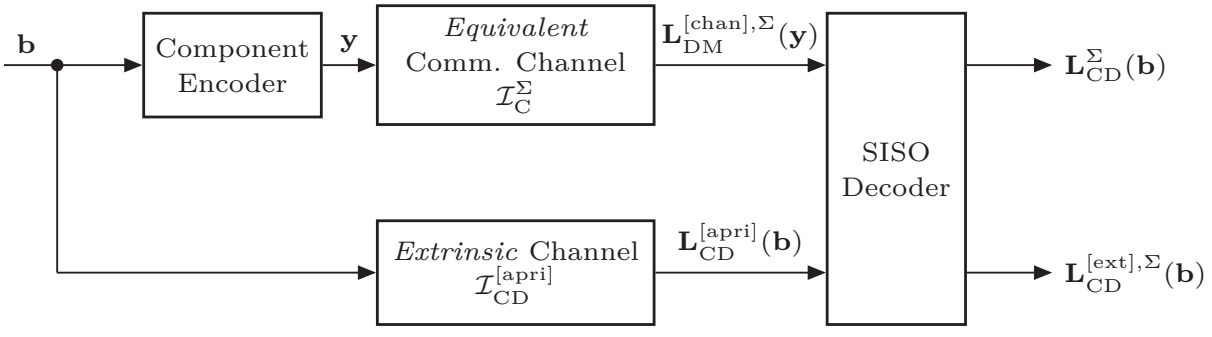


Figure 4.7: Information theoretic transmission model for RCRC codes.

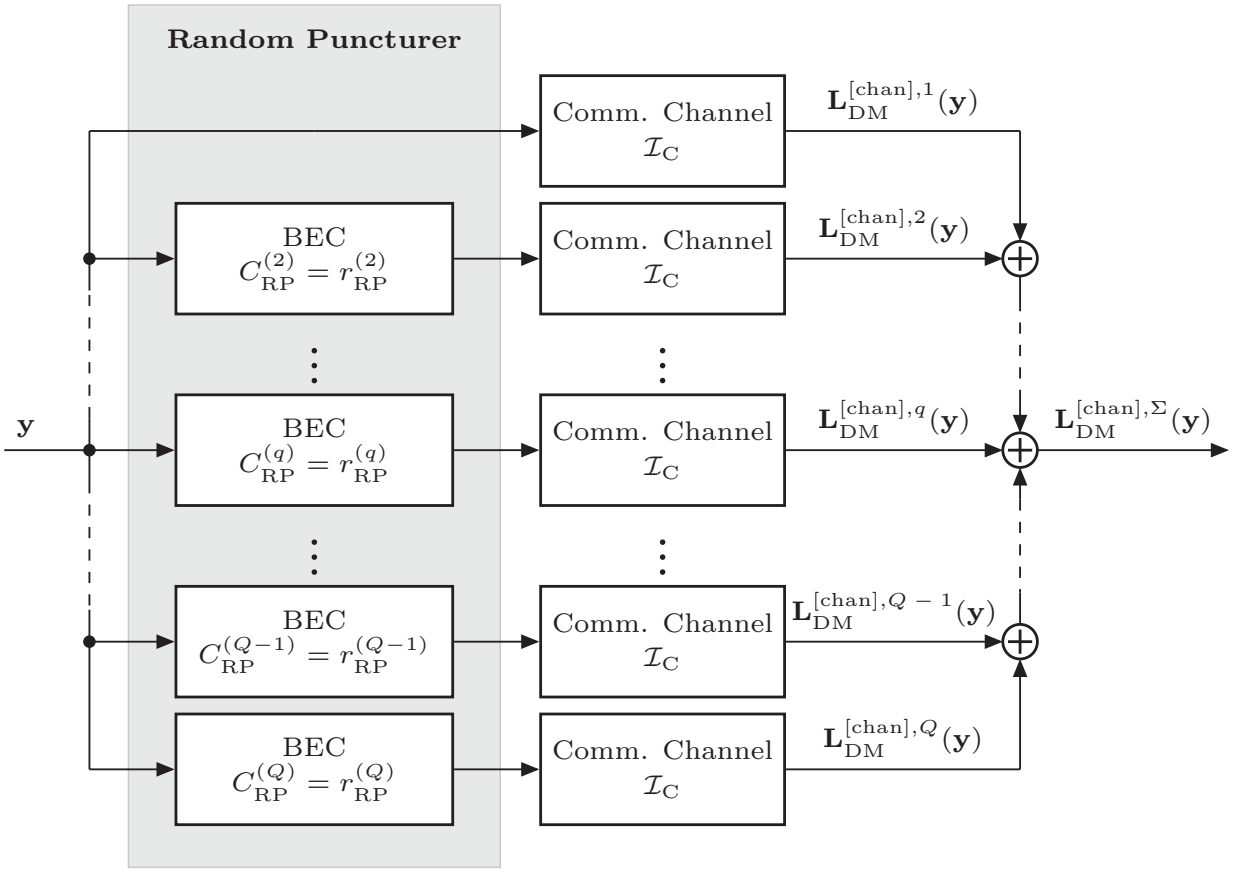


Figure 4.8: *Equivalent* communication channel.

the output of the *equivalent* communication channel. Due to the information defect[2] [LHHH05], a simple addition of all partial mutual information at the output of each branch is not possible. However, expressions for the mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ at the output of an arbitrary number of parallel concatenated BECs and BI-AWGNCs, respectively, can be derived.

Parallel Concatenation of Q BECs If the communication channel is a BEC with erasure probability P_{C} and mutual information $\mathcal{I}_{\mathrm{C}} = 1 - P_{\mathrm{C}}$, then the mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ provided by the considered RCRC code is given by the following lemma.

Lemma 4.2.1 *Consider the parallel concatenation of $Q = r_{\mathrm{DR}}^{-1}$ independent and memoryless BECs according to Figure 4.8. Let each channel q ($1 \leq q \leq Q$) be a serial concatenation of two BECs with overall mutual information*

$$\mathcal{I}_{\mathrm{C}}^{(q)} = r_{\mathrm{RP}}^{(q)} \cdot \mathcal{I}_{\mathrm{C}}. \tag{4.20}$$

[2]For the example of two parallel concatenated channels, the information defect denotes the fraction of information regarding an input bit which is delivered jointly by both channels. This information has to be subtracted once.

Then, the mutual information at the output of the parallel concatenated channels can be expressed as

$$\mathcal{I}_{\mathrm{C}}^{\Sigma} = 1 - \prod_{q=1}^{Q} \left(1 - \mathcal{I}_{\mathrm{C}}^{(q)}\right) = 1 - \prod_{q=1}^{Q} \left(1 - r_{\mathrm{RP}}^{(q)} \cdot \mathcal{I}_{\mathrm{C}}\right). \tag{4.21}$$

Proof 4.2.1 *See Appendix B.*

From Lemma 4.2.1 it follows that the overall mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ is dependent on the partial puncturing rates $r_{\mathrm{RP}}^{(q)}$. Consequently, optimal random repetition is achieved using optimal puncturing rates $r_{\mathrm{RP}}^{(q)}$ if the mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ at the output of the parallel concatenated channels is maximized.

Lemma 4.2.2 *The mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ between the code bits $\mathbf{y}$ and the corresponding LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}],\Sigma}(\mathbf{y})$ at the output of Q parallel concatenated BECs, each with mutual information $\mathcal{I}_{\mathrm{C}}^{(q)} = r_{\mathrm{RP}}^{(q)} \cdot \mathcal{I}_{\mathrm{C}}$ $(1 \leq q \leq Q)$, is maximized by the puncturing rates*

$$r_{\mathrm{RP}}^{(q)} = \begin{cases} 1 & , \quad 1 \leq q < Q' \\ Q \cdot r_{\mathrm{RP}} - (Q' - 1) & , \quad q = Q' \\ 0 & , \quad Q' < q \leq Q \end{cases} \quad , \quad Q' = \lceil Q \cdot r_{\mathrm{RP}} \rceil, \tag{4.22}$$

subject to

$$r_{\mathrm{RP}} = \frac{1}{Q} \sum_{q=1}^{Q} r_{\mathrm{RP}}^{(q)}, \quad r_{\mathrm{RP}}^{(1)} = 1, \tag{4.23}$$

i.e., under the constraint of an equal average puncturing rate r_{RP}.

Proof 4.2.2 *See Appendix B.*

Lemma 4.2.2 provides a tight upper bound on the mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ and declares that, first, $Q' = \lceil Q \cdot r_{\mathrm{RP}} \rceil = \lceil r_{\mathrm{R}}^{-1} \rceil$ memoryless and independent transmission channels are sufficient and, second, puncturing should only be employed within one branch. Without loss of generality, the branch Q' is selected throughout this contribution. Furthermore, the derivation in Appendix B provides a tight lower bound on $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ as mentioned below.

Corollary 4.2.3 *The mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ is lower bounded (worst case) by the puncturing rates*

$$r_{\mathrm{RP}}^{(1)} = 1, \tag{4.24}$$

$$\bar{r}_{\mathrm{RP}} = r_{\mathrm{RP}}^{(q)} = \frac{Q \cdot r_{\mathrm{RP}} - 1}{Q - 1} \quad , \quad 2 \leq q \leq Q, \tag{4.25}$$

i.e., for equal puncturing in each branch $q \geq 2$.

Proposition 4.2.4 *Using Lemma 4.2.1 and 4.2.2 as well as Corollary 4.2.3, $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ can be lower and upper bounded for an arbitrary puncturing scheme according to*

$$\mathcal{I}_{\mathrm{C}}^{\Sigma} \geq 1 - (1 - \mathcal{I}_{\mathrm{C}}) \cdot (1 - \bar{r}_{\mathrm{RP}} \cdot \mathcal{I}_{\mathrm{C}})^{Q-1} \tag{4.26}$$

$$\mathcal{I}_{\mathrm{C}}^{\Sigma} \leq 1 - (1 - \mathcal{I}_{\mathrm{C}})^{Q'-1} \cdot \left(1 - r_{\mathrm{RP}}^{(Q')} \cdot \mathcal{I}_{\mathrm{C}}\right), \tag{4.27}$$

with $r_{\mathrm{RP}}^{(Q')} = Q \cdot r_{\mathrm{RP}} - (Q' - 1)$.

It is further desirable to quantify the difference in terms of mutual information between the global minimum and the global maximum in order to evaluate the impact of the puncturing scheme. Note that the global maximum can be reached by optimal random repetition due to the tightness of the upper bound.

Considering (4.26), it can be concluded that the global minimum is given for $Q \to \infty$, assuming a fixed but arbitrary repetition rate $r_{\mathrm{R}} = 1/(Q \cdot r_{\mathrm{RP}})$ and a communication channel with fixed but arbitrary mutual information $\mathcal{I}_{\mathrm{C}}$:

$$\begin{aligned}
\mathcal{I}_{\mathrm{C,min}}^{\Sigma} &= \lim_{Q\to\infty} \left(1 - (1 - \mathcal{I}_{\mathrm{C}}) \cdot (1 - \bar{r}_{\mathrm{RP}} \cdot \mathcal{I}_{\mathrm{C}})^{Q-1}\right) \\
&= \lim_{Q\to\infty} \left(1 - (1 - \mathcal{I}_{\mathrm{C}}) \cdot \left(1 - \frac{Q \cdot r_{\mathrm{RP}} - 1}{Q - 1} \cdot \mathcal{I}_{\mathrm{C}}\right)^{Q-1}\right) \\
&= \lim_{Q\to\infty} \left(1 - (1 - \mathcal{I}_{\mathrm{C}}) \cdot \left(1 - \frac{r_{\mathrm{R}}^{-1} - 1}{Q - 1} \cdot \mathcal{I}_{\mathrm{C}}\right)^{Q-1}\right) \\
&= 1 - (1 - \mathcal{I}_{\mathrm{C}}) \cdot \exp\left(-\left(r_{\mathrm{R}}^{-1} - 1\right) \cdot \mathcal{I}_{\mathrm{C}}\right)
\end{aligned} \tag{4.28}$$

using the expression

$$\lim_{x\to\infty} \left(1 + \frac{a}{x}\right)^{x} = \exp(a). \tag{4.29}$$

The global maximum on $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ is independent of Q and given according to

$$\mathcal{I}_{\mathrm{C,max}}^{\Sigma} = 1 - (1 - \mathcal{I}_{\mathrm{C}})^{Q'-1} \cdot \left(1 - r_{\mathrm{RP}}^{(Q')} \cdot \mathcal{I}_{\mathrm{C}}\right). \tag{4.30}$$

The relative difference

$$\Delta\mathcal{I}_{\mathrm{C}}^{\Sigma} = \frac{\mathcal{I}_{\mathrm{C,max}}^{\Sigma} - \mathcal{I}_{\mathrm{C,min}}^{\Sigma}}{\mathcal{I}_{\mathrm{C,min}}^{\Sigma}} \tag{4.31}$$

is plotted in Figure 4.9 in percent against the repetition rate $r_{\mathrm{R}} = 1/(Q \cdot r_{\mathrm{RP}})$ for communication channels with different mutual information $\mathcal{I}_{\mathrm{C}} \in \{0.1, 0.3, 0.5\}$. For the considered channel conditions, gains up to 8 % in terms of mutual information are obtained by the optimal random repetition scheme compared to the lower bound given in (4.28).

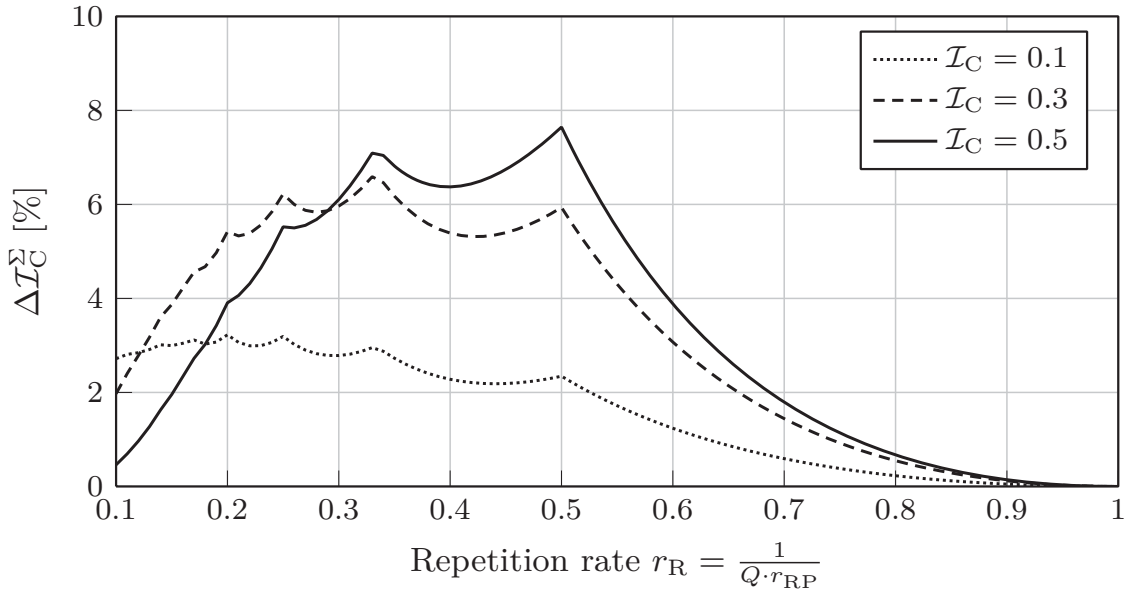


Figure 4.9: Gain in terms of channel-related mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$.

The relative difference $\Delta\mathcal{I}_{\mathrm{C}}^{\Sigma}$ is further illustrated in Figure 4.10 (left hand side) for the special condition $Q = Q'$ yielding $(Q'-1)/Q' < r_{\mathrm{RP}} \leq 1$. In this case, optimal/worst case puncturing provides the maximum/minimum channel-related mutual information of

$$\mathcal{I}_{\mathrm{C,max}}^{\Sigma} = 1 - (1 - \mathcal{I}_{\mathrm{C}})^{Q'-1} \cdot \left(1 - r_{\mathrm{RP}}^{(Q')} \cdot \mathcal{I}_{\mathrm{C}}\right) \tag{4.32}$$

$$\mathcal{I}_{\mathrm{C,min}}^{\Sigma} = 1 - (1 - \mathcal{I}_{\mathrm{C}}) \cdot (1 - \bar{r}_{\mathrm{RP}} \cdot \mathcal{I}_{\mathrm{C}})^{Q'-1}. \tag{4.33}$$

At repetition rates of $r_{\mathrm{R}} = 1/q$, $q \in \mathbb{N}$, the effect of the puncturing disappears and $\Delta\mathcal{I}_{\mathrm{C}}^{\Sigma}$ is identical to zero. However, between these rates optimal puncturing provides an observable gain. This gain increases with a decreasing puncturing rate r_{RP} and, furthermore, an increasing mutual information $\mathcal{I}_{\mathrm{C}}$.

However, the *extrinsic* information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{ext}]}$ generated by the SISO decoder rather than $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ itself is essential for the iterative decoding performance. This information depends on the employed convolutional mother code and the mutual information $\mathcal{I}_{\mathrm{C}}$ of the communication channel.

Figure 4.10 (right hand side) shows exemplarily the EXIT functions of the worst case puncturing (dashed lines) and optimal puncturing (bold lines) for the LTE component code with octal generator polynomials $\mathbf{G}_{\mathrm{CC}} = (1, 13/15)_8$ and constraint length $\mathfrak{L} + 1 = 4$ at different channel qualities $\mathcal{I}_{\mathrm{C}} \in \{0.1, 0.3, 0.5\}$ and a repetition rate $r_{\mathrm{R}} = 0.49$ (point ① in Figure 4.10 (left hand side)). Only for moderate channel qualities (e.g. $\mathcal{I}_{\mathrm{C}} = 0.3$), an increase in terms of *extrinsic* information is visible while for bad (e.g. $\mathcal{I}_{\mathrm{C}} = 0.1$) and good (e.g. $\mathcal{I}_{\mathrm{C}} = 0.5$) channel qualities there is no notable difference. It is worth mentioning that these conclusions are only valid for the considered LTE component code and the special

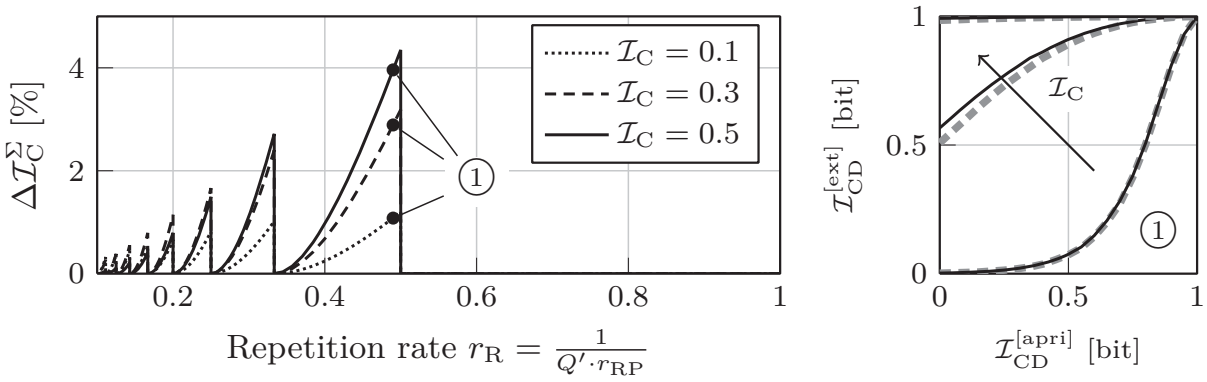


Figure 4.10: Left: Gain in terms of channel-related mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ for the special case $Q = Q'$. Right: EXIT charts for $\mathbf{G}_{\mathrm{CC}} = (1, 13/15)_8$ at point of interest ($r_{\mathrm{R}} = 0.49$) for channel qualities $\mathcal{I}_{\mathrm{C}} \in \{0.1, 0.3, 0.5\}$. Optimal puncturing depicted by solid lines and worse case puncturing by dashed lines.

case $Q = Q'$. For $Q > Q'$, the gains will be increased due to the increased relative difference $\Delta\mathcal{I}_{\mathrm{C}}^{\Sigma}$.

Parallel Concatenation of Q BIAWGNCs In the BIAWGNC case, the mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ provided by the RCRC code can be expressed by means of ten Brink's J-function according to [BR05]:

$$\mathcal{I}_{\mathrm{C}}^{\Sigma} = J\left(\sigma_{\mathcal{L}}^{\Sigma}\right), \tag{4.34}$$

where $(\sigma_{\mathcal{L}}^{\Sigma})^2$ denotes the variance of the LLRs $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}],\Sigma}(\mathbf{y})$ at the output of the *equivalent* communication channel. For repetition codes it is well-known that equal decoding performance can be achieved by transmitting the information bits without redundancy but with the same overall energy, i.e., $E_b^{\Sigma} = E_b$. This implies that the energy per modulation symbol E_s is increased by $r_{\mathrm{DR}}^{-1} = Q$ resulting in

$$E_s^{\Sigma} = r_{\mathrm{DR}}^{-1} \cdot E_s. \tag{4.35}$$

Assuming $E_s = 1$ and denoting the variance of the additive and Gaussian distributed noise signal by $\sigma_n^2 = N_0/2$ ($N_0/2$ is the spectral noise power density of the communication channel), this can also be interpreted as a transmission of the code bit vector $\mathbf{y}$ over a BIAWGNC with decreased spectral noise power density $(\sigma_n^{\Sigma})^2 = N_0^{\Sigma}/2 = N_0/(2 \cdot E_s^{\Sigma})$:

$$\frac{E_s^{\Sigma}}{N_0} = r_{\mathrm{DR}}^{-1} \cdot \frac{E_s}{N_0} \tag{4.36}$$

$$\Leftrightarrow \frac{1}{2\,(\sigma_n^{\Sigma})^2} = r_{\mathrm{DR}}^{-1} \cdot \frac{1}{2\,\sigma_n^2}. \tag{4.37}$$

According to (2.33), $(\sigma_{\mathcal{L}})^2 = 4/\sigma_n^2$ and $(\sigma_{\mathcal{L}}^{\Sigma})^2 = 4/(\sigma_n^{\Sigma})^2$ denote the variances of the LLRs at the output of the communication channel and the *equivalent* communication channel, respectively. Hence, (4.37) can be converted to

$$\begin{aligned} &\Leftrightarrow \frac{\left(\sigma_{\mathcal{L}}^{\Sigma}\right)^2}{8} = r_{\mathrm{DR}}^{-1} \cdot \frac{\sigma_{\mathcal{L}}^2}{8} \\ &\Leftrightarrow \sigma_{\mathcal{L}}^{\Sigma} \quad = \sqrt{r_{\mathrm{DR}}^{-1}} \cdot \sigma_{\mathcal{L}}. \end{aligned} \tag{4.38}$$

Neglecting the puncturing at first (i.e. $r_{\mathrm{RP}} = 1$), (4.38) delivers directly the desired relation

$$J\left(\sqrt{r_{\mathrm{DR}}} \cdot \sigma_{\mathcal{L}}^{\Sigma}\right) = J\left(\sigma_{\mathcal{L}}\right) \tag{4.39}$$

between the variances of the channel-related LLRs of each branch and the overall LLRs at the output of the parallel concatenated channels. $J\left(\sigma_{\mathcal{L}}\right)$ corresponds to the mutual information of the communication channel in each branch q according to Figure 4.8. The puncturing with overall rate r_{RP} causes a linear reduction by the factor r_{RP} of the mutual information $J\left(\sigma_{\mathcal{L}}\right)$. Consequently, it can be incorporated in (4.39) as a multiplicative factor [BR05]:

$$J\left(\sqrt{r_{\mathrm{DR}}} \cdot \sigma_{\mathcal{L}}^{\Sigma}\right) = r_{\mathrm{RP}} \cdot J\left(\sigma_{\mathcal{L}}\right) \tag{4.40}$$

$$\Leftrightarrow \sigma_{\mathcal{L}}^{\Sigma} \quad = \frac{1}{\sqrt{r_{\mathrm{DR}}}} \cdot J^{-1}\left(r_{\mathrm{RP}} \cdot J\left(\sigma_{\mathcal{L}}\right)\right). \tag{4.41}$$

EXIT functions The expressions for the channel-related mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ derived for the BEC and the BIAWGNC case are used for the prediction of the behavior of the SISO decoder. The RCRC decoder employs the SISO decoder which is fully determined by the employed convolutional mother code. Consequently, it generates equal *a posteriori* and *extrinsic* LLRs if assuming equal mutual input information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ and $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}$ and if both input LLR vectors $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}],\Sigma}(\mathbf{y})$ and $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{y})$ follow the same probability distribution. The latter condition is only fulfilled for the BEC. In this case, the EXIT function $\mathcal{T}_{\mathrm{CD}}^{(\mathrm{RC})}$ of the RCRC code can be constructed from the EXIT function $\mathcal{T}_{\mathrm{CD}}^{(\mathrm{CC})}$ of the employed convolutional mother code according to

$$\mathcal{T}_{\mathrm{CD}}^{(\mathrm{RC})}\left(\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}\middle|\mathcal{I}_{\mathrm{C}}\right) = \mathcal{T}_{\mathrm{CD}}^{(\mathrm{CC})}\left(\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}\middle|\mathcal{I}_{\mathrm{C}}^{\Sigma}\right), \tag{4.42}$$

with $0 \leq \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} \leq 1$, $0 \leq \mathcal{I}_{\mathrm{C}} \leq 1$, $0 \leq \mathcal{I}_{\mathrm{C}}^{\Sigma} \leq 1$, and $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ given by (4.27) and (4.34), respectively. The shift in the operation point from $(\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}, \mathcal{I}_{\mathrm{C}})$ to $(\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}, \mathcal{I}_{\mathrm{C}}^{\Sigma})$ can be interpreted as a transmission without repetition coding over a communication channel with increased mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$. Even though (4.42) is only valid for a communication channel modeled as BEC, it will be shown in the following example that it also provides a close approximation for the BIAWGNC case.

In order to obtain a computation rule for the *equivalent* erasure probability P_{C}^{Σ} and the *equivalent* channel SNR E_s^{Σ}/N_0, (4.6) and (4.9) have to be transformed with respect to (2.33), (2.34), (4.27), and (4.41) according to

$$P_{\mathrm{C}}^{\Sigma} = 1 - \mathcal{I}_{\mathrm{C}}^{\Sigma} = (1 - \mathcal{I}_{\mathrm{C}})^{Q'-1} \cdot \left(1 - r_{\mathrm{RP}}^{(Q')} \cdot \mathcal{I}_{\mathrm{C}}\right) \tag{4.43}$$

$$\frac{E_s^{\Sigma}}{N_0} = \frac{1}{8 \cdot r_{\mathrm{DR}}} \cdot \left[J^{-1}\left(r_{\mathrm{RP}} \cdot J\left(2 \cdot \sqrt{2 \cdot \frac{E_s}{N_0}} \right) \right) \right]^2 . \tag{4.44}$$

Example

The EXIT function of an RCRC code is determined according to (4.42) for the following scenarios:

(a) Rate-1/2 component code used in LTE, communication channel: BEC with erasure probability $P_{\mathrm{C}} = 0.8$, *extrinsic* channel: BEC,

(b) Rate-1/2 component code used in LTE, communication channel: BIAWGNC with $E_s/N_0 = -6\,\mathrm{dB}$, *extrinsic* channel: BIAWGNC.

The EXIT charts are depicted in Figure 4.11 (bold lines) and compared with their measured versions (dashed lines) for target code rates after rate matching of $r_{\mathrm{RC}} = \{1/3, 1/4, 1/5, 1/6\}$. As expected, the derived EXIT functions exactly match their measured versions for case (a) and provide a close approximation for case (b).

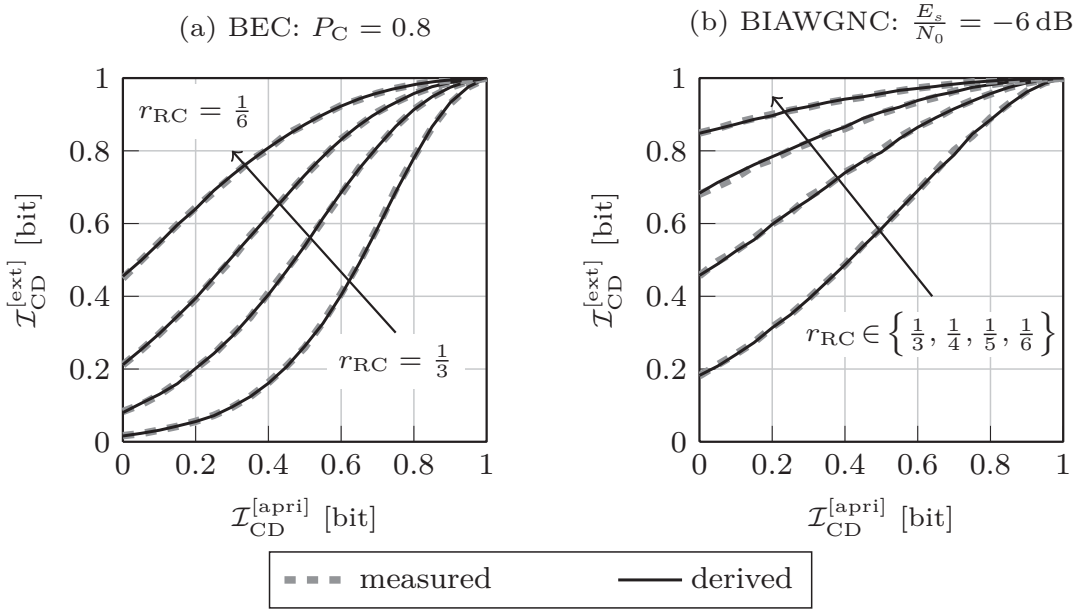


Figure 4.11: Comparison of derived (bold lines) and measured EXIT functions (dashed lines) for the rate-1/2 LTE component code.

Serial Concatenation

The expressions derived for PCCCs can directly be applied to inner codes of SCCCs by adapting the notation accordingly. Consequently, the EXIT function $\mathcal{T}_{\text{in}}^{(\text{RC})}$ of the inner RCRC code is determined, based on the EXIT function $\mathcal{T}_{\text{in}}^{(\text{CC})}$ of the underlying convolutional mother code, by

$$\mathcal{T}_{\text{in}}^{(\text{RC})}\left(\mathcal{I}_{\text{in}}^{[\text{apri}]}\middle|\,\mathcal{I}_{\text{C}}\right)=\mathcal{T}_{\text{in}}^{(\text{CC})}\left(\mathcal{I}_{\text{in}}^{[\text{apri}]}\middle|\,\mathcal{I}_{\text{C}}^{\Sigma}\right). \tag{4.45}$$

In (4.45), $\mathcal{I}_{\text{C}}^{\Sigma}$ is given according to (4.27) and (4.34), respectively.

For an RCRC code used as outer code in an SCCC scheme, the *extrinsic* channel with mutual information $\mathcal{I}_{\text{out}}^{[\text{apri}]}$ (see Figure 2.9) is substituted by an *equivalent extrinsic* channel with mutual information $\mathcal{I}_{\text{out}}^{[\text{apri}],\Sigma}$, which resembles the structure of the channel model illustrated in Figure 4.8.

Then, the maximum *a priori* information for the BEC case can be obtained by Proposition 4.2.4:

$$\mathcal{I}_{\text{out}}^{[\text{apri}],\Sigma}=1-\left(1-\mathcal{I}_{\text{out}}^{[\text{apri}]}\right)^{Q'-1}\cdot\left(1-r_{\text{RP}}^{(Q')}\cdot\mathcal{I}_{\text{out}}^{[\text{apri}]}\right). \tag{4.46}$$

For the BIAWGNC case, this information is given in accordance with (4.34) and (4.41) by

$$\mathcal{I}_{\text{out}}^{[\text{apri}],\Sigma}=J\left(\sigma_{\mathcal{L}}^{\Sigma}\right), \tag{4.47}$$

$$\sigma_{\mathcal{L}}^{\Sigma}=\frac{1}{\sqrt{r_{\text{DR}}}}\cdot J^{-1}\left(r_{\text{RP}}\cdot J\left(\sigma_{\mathcal{L}}\right)\right), \tag{4.48}$$

where $(\sigma_{\mathcal{L}})^2$ and $(\sigma_{\mathcal{L}}^{\Sigma})^2$ denote the variances of the *a priori* LLRs $\mathbf{L}_{\text{out}}^{[\text{apri}]}(\mathbf{e})$ and $\mathbf{L}_{\text{out}}^{[\text{apri}],\Sigma}(\mathbf{e})$. The latter LLR vector contains the output LLRs of the *equivalent extrinsic* channel.

The EXIT function $\mathcal{T}_{\text{out}}^{(\text{RC})}$ of the outer code in a serially concatenated RCRC code can then be constructed from the EXIT function $\mathcal{T}_{\text{out}}^{(\text{CC})}$ of the employed convolutional mother code according to

$$\mathcal{T}_{\text{out}}^{(\text{RC})}\left(\mathcal{I}_{\text{out}}^{[\text{apri}]}\right)=\mathcal{T}_{\text{out}}^{(\text{CC})}\left(\mathcal{I}_{\text{out}}^{[\text{apri}],\Sigma}\right) \tag{4.49}$$

with $0\leq\mathcal{I}_{\text{out}}^{[\text{apri}]}\leq 1$, $0\leq\mathcal{I}_{\text{out}}^{[\text{apri}],\Sigma}\leq 1$, and $\mathcal{I}_{\text{out}}^{[\text{apri}],\Sigma}$ given by (4.46) and (4.47), respectively. Again, equality in (4.45) and (4.49) is only given for the BEC case.

4.2.2 Optimization by EXIT Charts

An EXIT chart optimized rate matching procedure for random repetition is developed in this section. The minimum repetition rate $r_{\text{R}}^{\min}$ which allows for near error-free decoding at a specific channel quality P_{C} or E_s/N_0 is determined taking

Algorithm 4.2 RCRC Turbo codes: Optimization by EXIT charts

Aim: *Determine the minimum repetition rate r_R^{min} required for near error-free transmission at a given channel quality E_s/N_0 and with a given maximum number of Turbo iterations.*

(1) Determine by means of simulations the channel quality at which near error-free transmission becomes possible for the employed mother Turbo code, taking the number of Turbo iterations into account.

This channel quality corresponds to the channel quality P_{C}^{Σ} or E_s^{Σ}/N_0 of the *equivalent* communication channel (Figure 4.8).

(2) Transform P_{C}^{Σ} or E_s^{Σ}/N_0 to the mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ of the *equivalent* communication channel after rate matching according to (4.43) and (4.34), respectively:

$$\mathcal{I}_{\mathrm{C}}^{\Sigma} = \begin{cases} 1 - P_{\mathrm{C}}^{\Sigma} & : \text{ BEC} \\ J\left(2 \cdot \sqrt{2 \cdot \frac{E_s^{\Sigma}}{N_0}}\right) & : \text{ BIAWGNC} \end{cases} .$$

(3) Similarly, compute the mutual information $\mathcal{I}_{\mathrm{C}}$ of the communication channel for the target channel qualities P_{C} or E_s/N_0:

$$\mathcal{I}_{\mathrm{C}} = \begin{cases} 1 - P_{\mathrm{C}} & : \text{ BEC} \\ J\left(2 \cdot \sqrt{2 \cdot \frac{E_s}{N_0}}\right) & : \text{ BIAWGNC} \end{cases} .$$

(4) Compute the repetition rate $r_{\mathrm{R}}^{\min} = r_{\mathrm{DR}}^{\min}/r_{\mathrm{RP}}^{\min}$ which is required to obtain the mutual information of the *equivalent* communication channel $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ for the given target channel quality P_{C} or E_s/N_0 using (4.43) and (4.44), respectively. This repetition rate is determined by the parameter tuple $r_{\mathrm{DR}}^{\min} = 1/Q'_{\min}$ and $r_{\mathrm{RP}}^{\min}$:

$$\mathcal{I}_{\mathrm{C}}^{\Sigma} = \begin{cases} 1 - (1 - \mathcal{I}_{\mathrm{C}})^{Q'_{\min}-1} \cdot \left(1 - r_{\mathrm{RP}}^{(Q'_{\min})} \cdot \mathcal{I}_{\mathrm{C}}\right) & : \text{ BEC} \\ J\left(\frac{1}{\sqrt{r_{\mathrm{DR}}^{\min}}} \cdot J^{-1}\left(r_{\mathrm{RP}}^{\min} \cdot J\left(\sigma_{\mathcal{L}}\right)\right)\right) & : \text{ BIAWGNC} \end{cases} .$$

The parameter tuples $(Q'_{\min}, r_{\mathrm{RP}}^{(Q'_{\min})})$ and $(r_{\mathrm{DR}}^{\min}, r_{\mathrm{RP}}^{\min})$, respectively, which solves the above equation has to be determined numerically.

the number of Turbo iterations into account. The semi-analytical procedure is similar to that one proposed in Section 4.1.2 for rate matching by means of bit puncturing. A summary of the procedure is given in Algorithm 4.2.

As an example, the parallel concatenated, systematic rate-1/3 LTE Turbo code is employed as mother code. Near error-free transmission shall be performed over a BIAWGNC at $E_s/N_0 = -6\,\mathrm{dB}$ with a maximum of 10 Turbo iterations. In order to optimize the spectral efficiency of the considered system, the minimum repetition rate $r_{\mathrm{R}}^{\mathrm{min}}$ is determined following Algorithm 4.2. In a first step, Monte-Carlo simulations reveal that the LTE mother Turbo code has the desired EXIT chart characteristic for a channel quality of $E_s^{\Sigma}/N_0 = -4.8\,\mathrm{dB}$. The remaining objective is now to find analytically a repetition rate yielding an EXIT chart characteristic for the RCRC Turbo code at $E_s/N_0 = -6\,\mathrm{dB}$ which is identical to the EXIT chart characteristic of the mother Turbo code. Executing step (2) - (4) results in a minimum repetition rate of $r_{\mathrm{R}}^{\mathrm{min}} \approx 0.7$ ($Q'_{\mathrm{min}} = 2$, $r_{\mathrm{RP}}^{\mathrm{min}} \approx 0.7$) and, thus, in an effective code rate after Turbo coding and rate matching of $r_{\mathrm{PCCC}} \approx 0.23$.

The EXIT chart of the optimized RCRC Turbo code (bold lines) and the decoding trajectory (dashed line) are depicted in Figure 4.12 (left hand side). Approximately 10 Turbo iterations are sufficient for near error-free decoding which complies with the corresponding BER depicted in Figure 4.12 as well. To obtain the BER of the RCRC Turbo code, a transmission of $\ell_b = 100000$ information bits per frame has been simulated in an AWGN environment.

It can be concluded that also for RCRC codes, rate matching can be performed

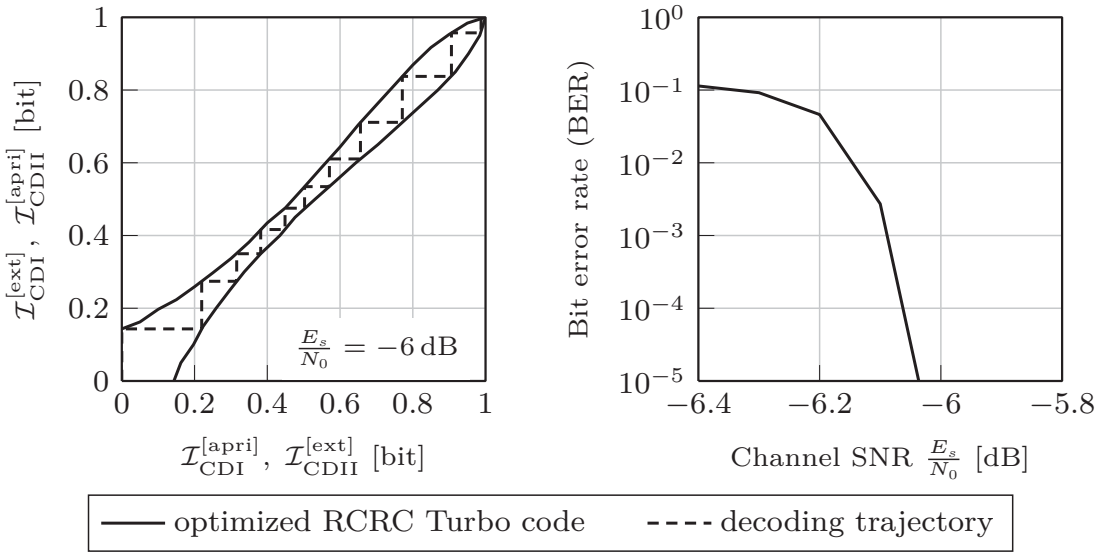


Figure 4.12: EXIT chart optimization at $E_s/N_0 = -6\,\mathrm{dB}$ for the systematic rate-1/3 LTE Turbo code. Left hand side: EXIT chart of the semi-analytically optimized RCRC Turbo code (bold lines) including its decoding trajectory (dashed line). Right hand side: corresponding BER performance.

on the fly based on the analytical expressions derived in the previous sections. Only the EXIT chart of the mother code has to be simulated once in advance.

4.2.3 Deterministic Repetition

It has been shown in Section 4.1.3 that the puncturing scheme specified in the LTE standard can be evaluated by a transmission model based on random processes with sufficient accuracy. This model has incorporated the possibility of different puncturing rates for the systematic and the parity fraction and, thus, has accounted for a specific property of the LTE puncturing scheme.

Considering the LTE bit repetition scheme, the transmission model of the previous section (see Figure 4.7) has to be adapted accordingly, resulting in the modified transmission model depicted in Figure 4.13. To account for a different repetition rate of the systematic bits $\mathbf{b}$ and the parity bits $\mathbf{c}$, they are individually transmitted over, in general, different *equivalent* communication channels with mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma,\mathrm{sys}}$ and $\mathcal{I}_{\mathrm{C}}^{\Sigma,\mathrm{par}}$ given according to (4.27) and (4.34), respectively. Furthermore, both channels are parameterized by their deterministic repetition rates $r_{\mathrm{DR}}^{\mathrm{sys}} = 1/Q_{\mathrm{sys}}$ and $r_{\mathrm{DR}}^{\mathrm{par}} = 1/Q_{\mathrm{par}}$ as well as their puncturing rates for the last branch $r_{\mathrm{RP}}^{(Q_{\mathrm{sys}})}$ and $r_{\mathrm{RP}}^{(Q_{\mathrm{par}})}$ (see Figure 4.8). The parity bits are generated by the non-systematic convolutional component encoder with octal generator polynomial(s) $\mathbf{G}_{\mathrm{nsys}}$ and code rate $r_{\mathrm{CC}}^{[\mathrm{nsys}]} = 1/(r_{\mathrm{CC}}^{-1} - 1)$. Together with the systematic branch, it represents the convolutional mother encoder with generator polynomial(s) $\mathbf{G}_{\mathrm{CC}} = (1, \mathbf{G}_{\mathrm{nsys}})$ and code rate r_{CC}. The parallel concatenation of the

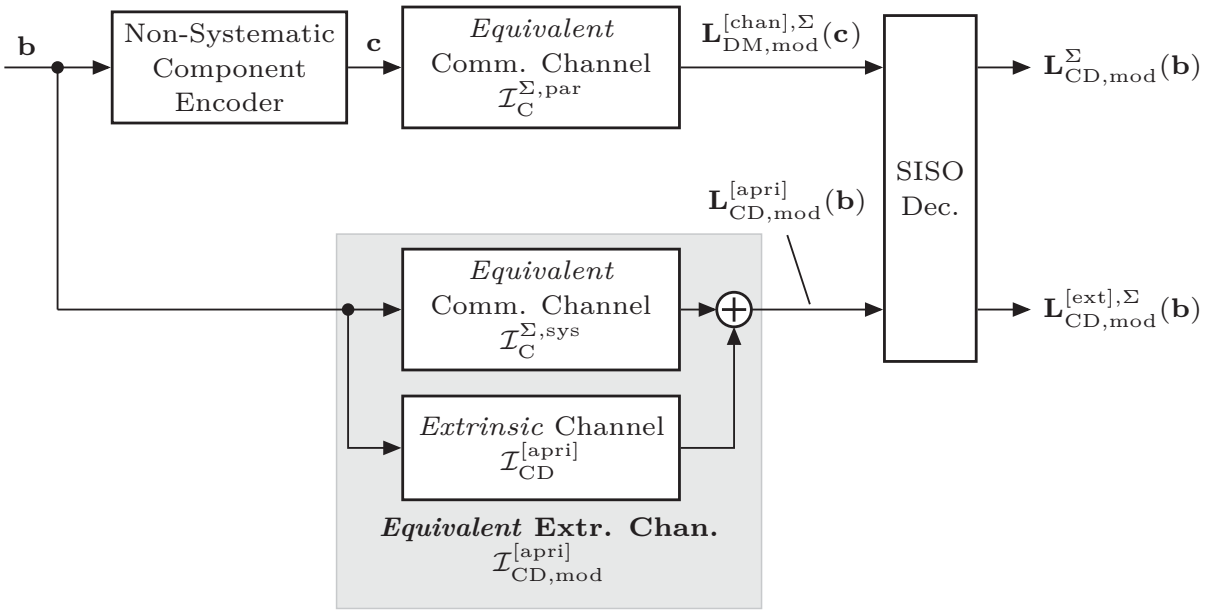


Figure 4.13: Modified information theoretic transmission model approximating the effect of the deterministic LTE bit repetition scheme.

equivalent communication channel with mutual information $\mathcal{I}_{\mathrm{C}}^{\Sigma,\mathrm{sys}}$ and the *extrinsic* channel with mutual information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}$ builds up an *equivalent extrinsic* channel with mutual information $\mathcal{I}_{\mathrm{CD,mod}}^{[\mathrm{apri}]}$.

The *equivalent* communication channel in the parity branch and the *equivalent extrinsic* channel deliver LLRs $\mathbf{L}_{\mathrm{DM,mod}}^{[\mathrm{chan}],\Sigma}(\mathbf{c})$ and $\mathbf{L}_{\mathrm{CD,mod}}^{[\mathrm{apri}]}(\mathbf{b})$, respectively, which are exploited by the SISO decoder to generate *extrinsic* LLRs $\mathbf{L}_{\mathrm{CD,mod}}^{[\mathrm{ext}],\Sigma}(\mathbf{b})$ and *a posteriori* LLRs $\mathbf{L}_{\mathrm{CD,mod}}^{\Sigma}(\mathbf{b})$. Obviously, the transmission model analyzed in the previous section is covered by the modified transmission model (special case $\mathcal{I}_{\mathrm{C}}^{\Sigma,\mathrm{sys}} = \mathcal{I}_{\mathrm{C}}^{\Sigma,\mathrm{par}}$).

In LTE, systematic bits and parity bits are repeated corresponding to the ring buffer implementation described in Section 3.4. Consequently, the repetition rate of the systematic bits and the repetition rate of the parity bits may be different. Modeling the communication channel and the *extrinsic* channel as a BEC, the EXIT function $\mathcal{T}_{\mathrm{CD,\,mod}}^{(\mathrm{RC})}$ of the novel RCRC code can be derived from the EXIT function $\mathcal{T}_{\mathrm{CD}}^{(\mathrm{nsys})}$ of the non-systematic component code as

$$\mathcal{T}_{\mathrm{CD,\,mod}}^{(\mathrm{RC})}\left(\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}\middle|\,\mathcal{I}_{\mathrm{C}}\right) = \mathcal{T}_{\mathrm{CD}}^{(\mathrm{nsys})}\left(\mathcal{I}_{\mathrm{CD,mod}}^{[\mathrm{apri}]}\middle|\,\mathcal{I}_{\mathrm{C,mod}}^{\Sigma}\right) \tag{4.50}$$

where $\mathcal{I}_{\mathrm{CD,mod}}^{[\mathrm{apri}]}$ and $\mathcal{I}_{\mathrm{C,mod}}^{\Sigma}$ are given by

$$\mathcal{I}_{\mathrm{CD,mod}}^{[\mathrm{apri}]} = \begin{cases} \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} + \mathcal{I}_{\mathrm{C}}^{\Sigma,\mathrm{sys}} \cdot \left(1 - \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}\right) & : \mathrm{BEC} \\ J\left(\sqrt{\left[J^{-1}\left(\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}\right)\right]^2 + \left[J^{-1}\left(\mathcal{I}_{\mathrm{C}}^{\Sigma,\mathrm{par}}\right)\right]^2}\right) & : \mathrm{BIAWGNC} \end{cases} \tag{4.51}$$

$$\mathcal{I}_{\mathrm{C,mod}}^{\Sigma} = \mathcal{I}_{\mathrm{C}}^{\Sigma,\mathrm{par}}. \tag{4.52}$$

Due to the constraint of equal probability distributions, this expression provides only an approximation for the BIAWGNC case.

The performance of the more flexible random repetition scheme described above and the deterministic repetition scheme employed in the LTE standard are compared in terms of their EXIT functions in Figure 4.14. The considered target code rates after rate-1/3 convolutional Turbo coding and bit repetition are chosen to $r_{\mathrm{PCCC}} \in \{1/4, 1/5, 1/6\}$ which can be determined by means of the parameters of the *equivalent* communication channels:

$$r_{\mathrm{PCCC}} = \frac{1}{3 + (Q_{\mathrm{sys}} - 1) \cdot r_{\mathrm{RP}}^{(Q_{\mathrm{sys}})} + 2 \cdot (Q_{\mathrm{par}} - 1) \cdot r_{\mathrm{RP}}^{(Q_{\mathrm{par}})}}. \tag{4.53}$$

Taking the block size $\ell_b = 6144$, the initial reading position $\theta_1 = 386$, and (4.53) into account, this results in the parameter set given in Table 4.1. For all considered target code rates, both *equivalent* communication channels are modeled by a parallel concatenation of two memoryless and independent communication

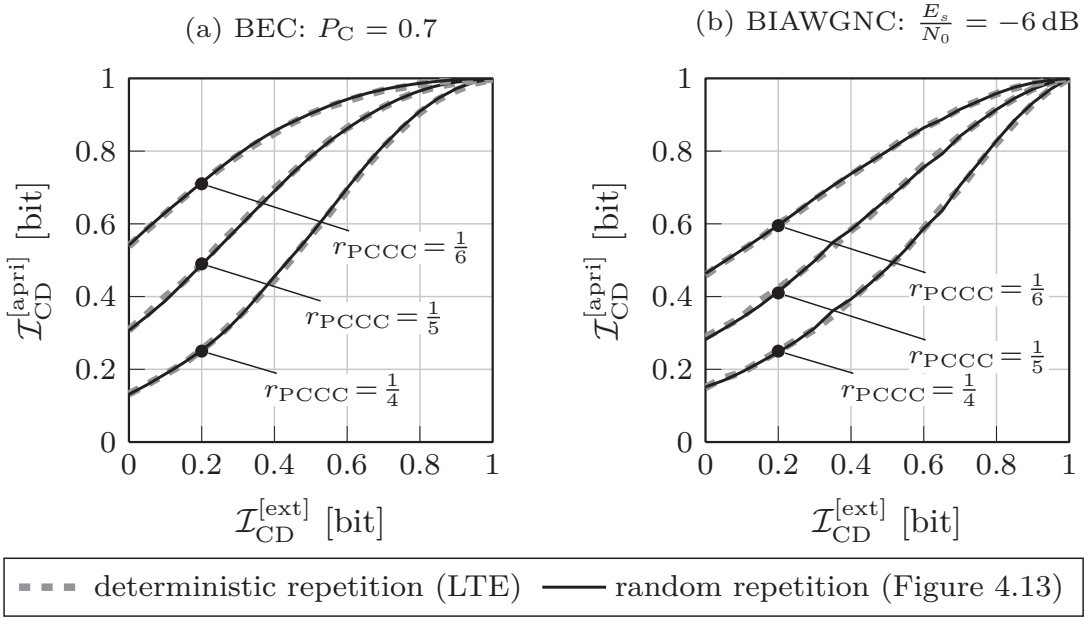


Figure 4.14: Random repetition according to Figure 4.13 versus deterministic repetition scheme specified in [3GP11] for the LTE component code.

channels, each with mutual information $\mathcal{I}_C$. Code rate adaptation is performed in this case only by changing the puncturing rates of the second (last) branch.

Transmission is exemplarily performed over a BEC with erasure probabilities $P_C = 0.7$ and over a BIAWGNC with $E_s/N_0 = -6\,\mathrm{dB}$. As illustrated in Figure 4.14, the deterministic LTE repetition scheme is accurately approximated by the random repetition scheme depicted in Figure 4.13. This enables the performance prediction and optimization of the standardized LTE system for rates $r_{PCCC} < 1/3$ reducing considerably the number of required Monte Carlo simulations.

4.3 Rate-Compatible Insertion Convolutional Codes

Like RCRC codes, *Rate-Compatible Insertion Convolutional* (RCIC) codes can be employed in communication systems in order to flexibly decrease the effective code rate before fixed rate convolutional (Turbo) coding. In what follows, their EXIT charts are theoretically derived in Section 4.3.1 for RCIC codes in which

Table 4.1: Parameter set used for the simulations depicted in Figure 4.14.

r_{PCCC}	Q_{par}	Q_{sys}	$r_{RP}^{(Q_{par})}$	$r_{RP}^{(Q_{sys})}$
1/4	2	2	0.032	0.937
1/5	2	2	0.532	0.937
1/6	2	2	1.000	1.000

dummy bits are randomly inserted. In Section 4.3.2, these results are then applied for the optimization of systems employing rate matching by means of random dummy bit insertion. Finally, they are compared with the EXIT charts obtained for systems applying the deterministic insertion rule proposed in Algorithm 3.1 (see Section 4.3.3).

4.3.1 Random Insertion

In order to derive an expression for the EXIT function of the insertion convolutional code by means of the EXIT function of the convolutional mother code, a random interleaver is assumed in Figure 3.6. In [BV11b], we have presented an information theoretic analysis of RCIC codes modeling the communication channel and the extrinsic feedback of the constituent decoder by BECs or BIAWGNCs. In what follows, this derivation will be extended to arbitrary communication/extrinsic channel models. Analytical expressions describing the influence of dummy bit insertion are given for PCCCs as well as for SCCCs.

Parallel Concatenation

The information theoretic transmission model for RCIC codes is depicted in Figure 4.15. The multiplexed vector $\mathbf{x} = [\mathbf{b}, \mathbf{d}]$ rather than the information bit vector $\mathbf{b}$ is fed into the convolutional component encoder since dummy bit insertion is employed *before* convolutional encoding. After convolutional encoding, the resulting bit stream $\mathbf{y}$ is transmitted over the communication channel with mutual information $\mathcal{I}_\mathrm{C}$, which provides LLRs $\mathbf{L}_\mathrm{DM}^{[\mathrm{chan}]}(\mathbf{y})$ to the SISO decoder. The *a priori* information $\mathbf{L}_\mathrm{CD}^{[\mathrm{apri}],\star}(\mathbf{x})$ which is also delivered to the SISO decoder consists of two fractions:

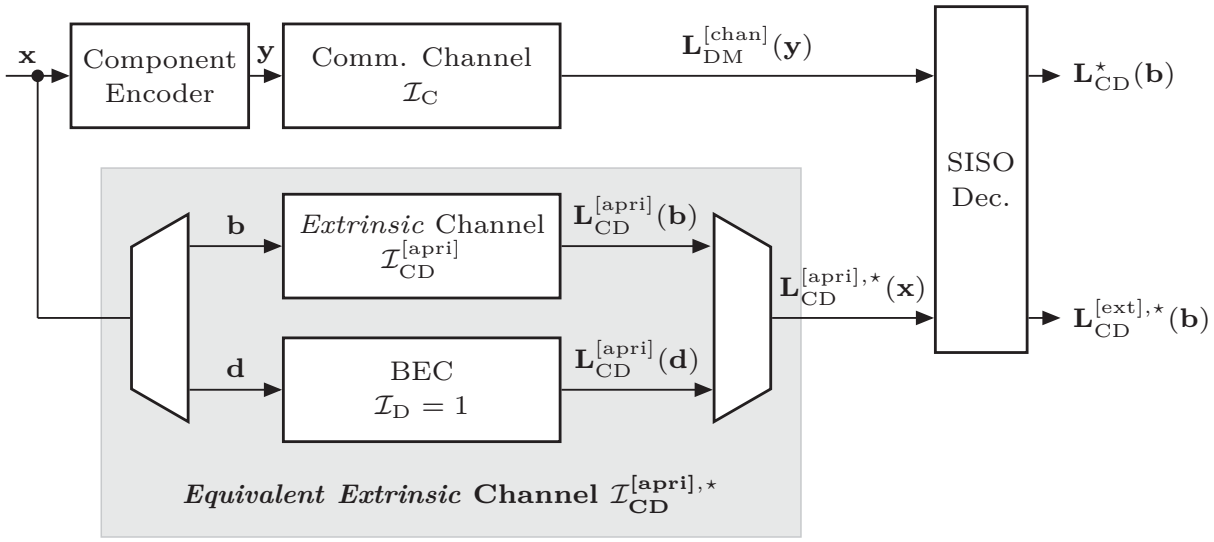


Figure 4.15: Information theoretic transmission model for RCIC codes.

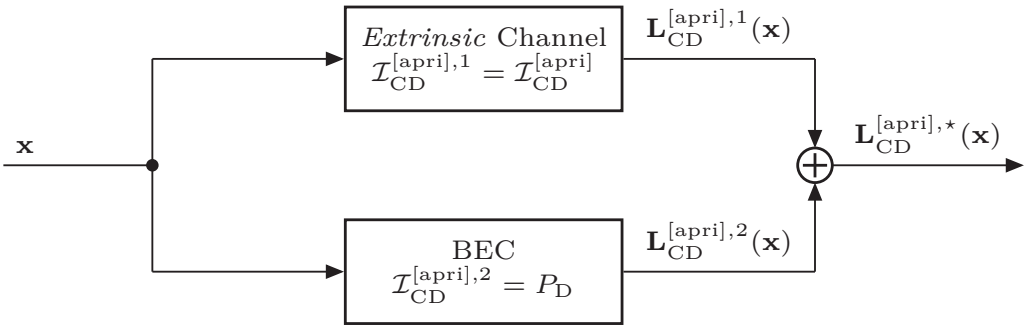


Figure 4.16: Transformed *equivalent extrinsic* channel.

1) The *a priori* LLRs $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}]}(\mathbf{b})$ of the information bits $\mathbf{b}$ provided by the other component decoder. This is modeled by the *extrinsic* channel with mutual information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}$.

2) The perfect *a priori* LLRs $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}]}(\mathbf{d}) \to \boldsymbol{\infty}$ provided by the dummy bits $\mathbf{d}$. This can be modeled by a BEC with mutual information $\mathcal{I}_{\mathrm{D}} = 1$.

If random interleaving is employed for the insertion of dummy bits, the *equivalent extrinsic* channel with mutual information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],\star}$, which is highlighted by the gray box in Figure 4.15, can be transformed into the structure shown in Figure 4.16 as follows:

The perfect mutual information on the dummy bits $\mathbf{d}$ within the vector $\mathbf{x}$ is modeled by a BEC with erasure probability

$$P_{\mathrm{E}} = 1 - P_{\mathrm{D}} = 1 - \frac{\ell_d}{\ell_x} = \frac{\ell_b}{\ell_x} \tag{4.54}$$

and mutual information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],2} = 1 - P_{\mathrm{E}} = P_{\mathrm{D}}$ according to (2.28). The BEC secures that only the dummy bit fraction $P_{\mathrm{D}} = \ell_d/\ell_x$ is exploited by the SISO decoder as perfect *a priori* information $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}]}(\mathbf{d}) \to \boldsymbol{\infty}$, while no additional information is generated for the information bit fraction, i.e., $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}]}(\mathbf{b}) = \mathbf{0}$. Note that due to the random nature of the BEC, only random insertion is covered by this model.

Furthermore, the *extrinsic* channel with mutual information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}$ delivers information on $\mathbf{b}$ and $\mathbf{d}$ which seems to be an improper modeling of the actual *extrinsic* channel. However, perfect information on the dummy bits has already been provided by the BEC so that no additional (new) information is provided by the *extrinsic* channel.

This results in the overall *a priori* LLR vector

$$\mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}],\star}(\mathbf{x}) = \mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}],1}(\mathbf{x}) + \mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}],2}(\mathbf{x}) \tag{4.55}$$

at the output of the parallel concatenated channels.

For the derivation of the EXIT function, the mutual Information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],\star}$ of the transformed *equivalent extrinsic* channel as shown in Figure 4.16 is required.

Proposition 4.3.1 *Consider a parallel concatenation of a BEC with mutual information* $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],2} = P_{\mathrm{D}}$ *and an arbitrary channel with mutual information* $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],1} = \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}$. *The overall mutual information* $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],\star}$ *of the parallel concatenated channel is then given by*

$$\begin{aligned}\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],\star} &= \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],1} + \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],2} - \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],1} \cdot \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],2} \\ &= \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} + P_{\mathrm{D}} \cdot \left(1 - \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}\right) \geq \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}. \end{aligned} \tag{4.56}$$

Proof 4.3.1 *See Appendix B.*

Consequently, dummy bits provide additional *a priori* information

$$\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],\star} - \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} = P_{\mathrm{D}} \cdot \left(1 - \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}\right)$$

to the SISO decoder resulting in a more reliable decoding of the information bits $\mathbf{b}$.

EXIT functions Equation (4.56) can now be used to determine the EXIT chart of the RCIC code with random dummy bit insertion based on the EXIT chart of the convolutional mother code. Due to the fact that the same SISO decoder is used in both systems, it can be concluded that equal *a posteriori* LLRs and *extrinsic* LLRs are generated for the same amount of input information if the *a priori* LLRs $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}],1}(\mathbf{x})$ and $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{apri}],\star}(\mathbf{x})$ follow the same probability distribution. However, the latter condition is only fulfilled for an *extrinsic* channel modeled as a BEC. Nevertheless close approximations can be obtained for other channel models as, e.g., a BIAWGNC.

Relying on the above considerations, the EXIT function $\mathcal{T}_{\mathrm{CD}}^{(\mathrm{IC})}$ of the RCIC code can be constructed according to

$$\mathcal{T}_{\mathrm{CD}}^{(\mathrm{IC})}\left(\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} \middle| \mathcal{I}_{\mathrm{C}}\right) = \mathcal{T}_{\mathrm{CD}}^{(\mathrm{CC})}\left(\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],\star} \middle| \mathcal{I}_{\mathrm{C}}\right), \tag{4.57}$$

with $\mathcal{T}_{\mathrm{CD}}^{(\mathrm{CC})}$ denoting the EXIT function of the underlying convolutional mother code and $0 \leq \mathcal{I}_{\mathrm{C}} \leq 1$, $0 \leq \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} \leq 1$, and $0 \leq \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],\star} \leq 1$. Obviously, the dummy bit insertion leads to a shift in the operation point towards higher *a priori* information, namely from $\left(\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}, \mathcal{I}_{\mathrm{C}}\right)$ to $\left(\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],\star}, \mathcal{I}_{\mathrm{C}}\right)$.

Example

As an example, the systematic rate-1/2 component code of the LTE Turbo encoder with generator polynomials $\mathbf{G}_{\mathrm{CC}} = (1, 15/13)_8$ and constraint length $\mathfrak{L} + 1 = 4$ is considered again.

The construction of the EXIT function after dummy bit insertion is visualized in Figure 4.17 for a transmission over a BEC with erasure probability $P_{\mathrm{C}} = 0.6$ and a dummy bit fraction of $P_{\mathrm{D}} = 0.5$. A fraction of 50 % dummy bits correspond to a mutual *a priori* information of $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],2} = 0.5$ which is initially exploited by the SISO decoder even if no *a priori* information about the information bits is given. The corresponding point can, thus, be mapped to the beginning of the RCIC EXIT function. According to (4.56), the remaining points are then shifted relatively to this point, e.g., the point at $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} = 0.75$ in the left plot is mapped to the value $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} = 0.5$ in the right plot and the point at $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} = 1$ to the identical value $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} = 1$. This leads to a stretching of the interval $[P_{\mathrm{D}} = 0.5, 1]$ to $[0, 1]$ as it is illustrated in Figure 4.17.

In order to validate the results, the EXIT functions of the resulting RCIC code are determined by means of (4.56) and (4.57) and compared to their measured versions in Figure 4.18 for different setups:

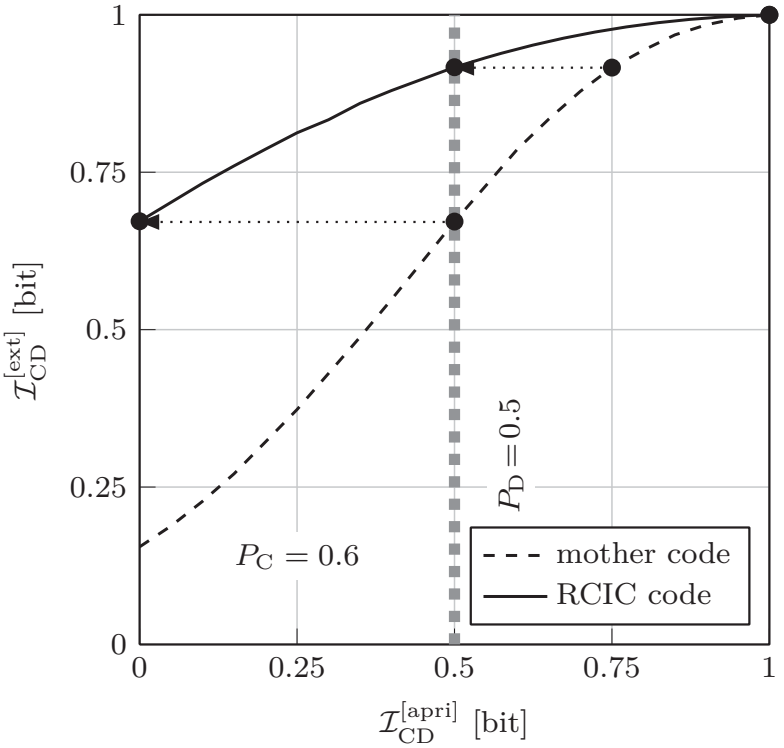


Figure 4.17: EXIT function of the RCIC code with 50 % dummy bit insertion constructed from the EXIT function of the convolutional mother code with $\mathbf{G}_{\mathrm{CC}} = (1, 15/13)_8$. Transmission has been performed over a BEC with $P_{\mathrm{C}} = 0.6$.

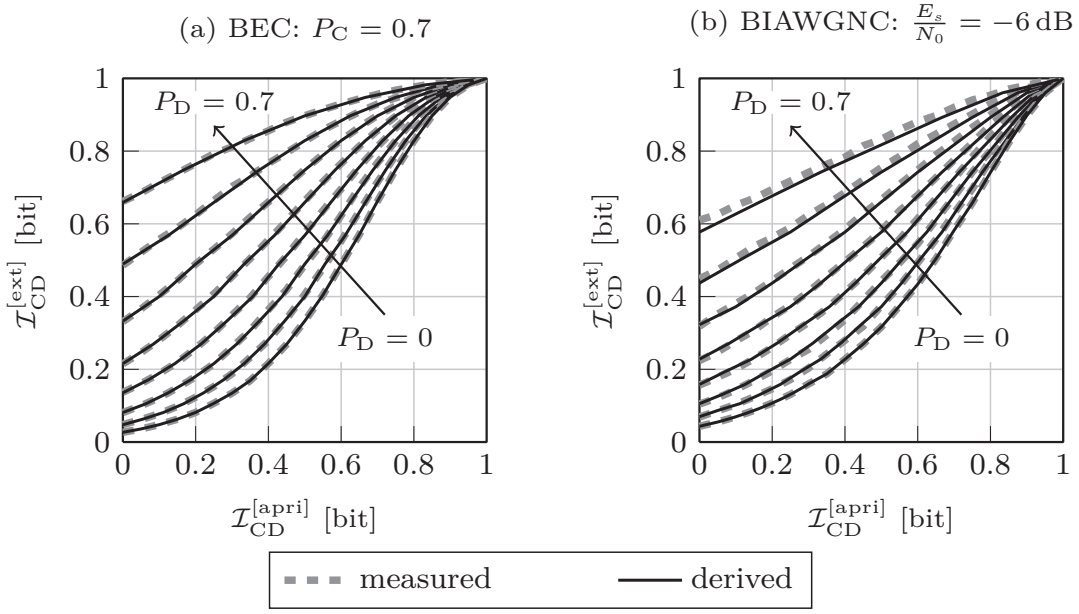


Figure 4.18: Comparison of derived (bold lines) and measured EXIT functions (dashed lines) for the LTE component code and different dummy bit fractions $P_D \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$.

(a) Rate-1/2 LTE component code, communication channel: BEC with erasure probability $P_C = 0.7$, *a priori* channel: BEC,

(b) Rate-1/2 LTE component code, communication channel: BIAWGNC with $E_s/N_0 = -6\,\mathrm{dB}$, *a priori* channel: BIAWGNC.

The derived EXIT functions (solid lines) and their measured versions (dashed lines) are plotted together in the same figure for different dummy bit fractions $P_D \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$.

It is shown in Figure 4.18 that, for the BEC case, the derived EXIT functions exactly match their measured versions, while for the BIAWGNC case, the derivation provides a very close approximation. As previously mentioned, the condition for the consistency of both curves is the equality of the amount of *a priori* information and the equality of the underlying probability distribution. However, the latter condition is not met for the BIAWGNC case due to the increasing divergence of both distributions for an increasing number of dummy bits. Accordingly, for a high fraction of dummy bits, the *a priori* LLRs $\mathbf{L}_{CD}^{[apri]}(\mathbf{x})$ at the output of the *equivalent extrinsic* channel cannot be modeled by a Gaussian distributed random process anymore. Nevertheless, the accuracy of this approximation increases rapidly with a decreasing number of inserted dummy bits. It is worth mentioning that both EXIT functions match almost perfectly for a dummy bit fraction of up to 40 % of the block size and, thus, for many relevant application scenarios.

Serial Concatenation

The expressions obtained for PCCCs can directly be adapted to inner codes of SCCCs by substituting the respective indices:

$$\mathcal{T}_{\text{in}}^{(\text{IC})}\left(\mathcal{I}_{\text{in}}^{[\text{apri}]}\middle|\,\mathcal{I}_{\text{C}}\right)=\mathcal{T}_{\text{in}}^{(\text{CC})}\left(\mathcal{I}_{\text{in}}^{[\text{apri}],\star}\middle|\,\mathcal{I}_{\text{C}}\right) \tag{4.58}$$

$$\mathcal{I}_{\text{in}}^{[\text{apri}],\star}=\mathcal{I}_{\text{in}}^{[\text{apri}]}+P_{\text{in}}\cdot\left(1-\mathcal{I}_{\text{in}}^{[\text{apri}]}\right)\geq\mathcal{I}_{\text{in}}^{[\text{apri}]} \tag{4.59}$$

$$P_{\text{in}}=\frac{\ell_d}{\ell_e+\ell_d}. \tag{4.60}$$

EXIT functions for outer SCCC codes cannot be determined based on the employed convolutional mother code, since the *a priori* LLRs of the input bits have to be adapted in this case. Monte-Carlo simulations have to be carried out instead.

4.3.2 Optimization by EXIT Charts

Rate matching by means of RCIC Turbo codes can also be optimized using EXIT charts. The minimum dummy bit fraction $P_{\text{D}}^{\min}$ required for near error-free decoding at a specific channel quality P_{C} or E_s/N_0 can be determined by means of the EXIT chart of the convolutional mother Turbo code. The semi-analytical rate matching procedure aims at minimizing P_{D}, $0\leq P_{\text{D}}<1$, subject to

$$\mathcal{T}_{\text{CD}}^{(\text{IC})}\left(\mathcal{I}_{\text{CD}}^{[\text{apri}]}\middle|\,\mathcal{I}_{\text{C}}\right)-\mathcal{I}_{\text{CD}}^{[\text{apri}]}>\Omega,\quad 0\leq\mathcal{I}_{\text{CD}}^{[\text{apri}]}\leq 1, \tag{4.61}$$

where Ω is a measure for the openness of the decoding tunnel and $\mathcal{T}_{\text{CD}}^{(\text{IC})}$ is given by (4.57). For $\Omega>0$, (4.61) secures that the EXIT curve lies above the main diagonal which prevents an intersection within the EXIT chart. The parameter Ω can be geometrically interpreted as the distance between one EXIT curve and the main diagonal on the ordinate. The optimization process is summarized in Algorithm 4.3.

As an example, the parallel concatenated, systematic rate-1/3 LTE Turbo code is chosen as mother code. Transmission shall be performed over a BIAWGNC with $E_s/N_0=-6\,\text{dB}$. The desired openness of the decoding tunnel is set to $\Omega=0.035$. At first, the EXIT chart of the mother code is simulated. Then, the dummy bit fraction P_{D} is continuously increased and the corresponding EXIT functions are computed using (4.56) and (4.57) until (4.61) is fulfilled. For the considered example, this results in a minimum dummy bit fraction of $P_{\text{D}}^{\min}=34.2\,\%$ and, thus, in an effective code rate after Turbo coding and rate matching of $r_{\text{PCCC}}\approx 0.248$.

The EXIT chart of the optimized RCIC code (bold lines), the decoding trajectory (dashed line) and the EXIT chart of the mother Turbo code (dotted lines) are depicted in Figure 4.19. Approximately 10 Turbo iterations are sufficient for near

Algorithm 4.3 RCIC Turbo codes: Optimization by EXIT charts

Aim: *Determine the minimum dummy bit fraction P_D required for near error-free transmission at a given channel quality E_s/N_0 and for a desired openness Ω.*

(1) Determine the EXIT chart of the mother code for the target channel quality P_C or E_s/N_0.

(2) Set Ω such that error-free decoding becomes possible for a moderate number of Turbo iterations. A small Ω leads to a system which requires many Turbo iterations to reach the top right corner of the EXIT charts.

(3) Compute the minimum dummy bit fraction $P_\mathrm{D}^{\min}$ which is required to fulfill (4.61) by means of (4.56) and (4.57).

error-free decoding. This is verified by the corresponding BER depicted in Figure 4.19 in which the bit error rate at the target channel quality $E_s/N_0 = -6\,\mathrm{dB}$ is marked by a circle. A frame size of $\ell_b = 100000$ has been selected for simulation.

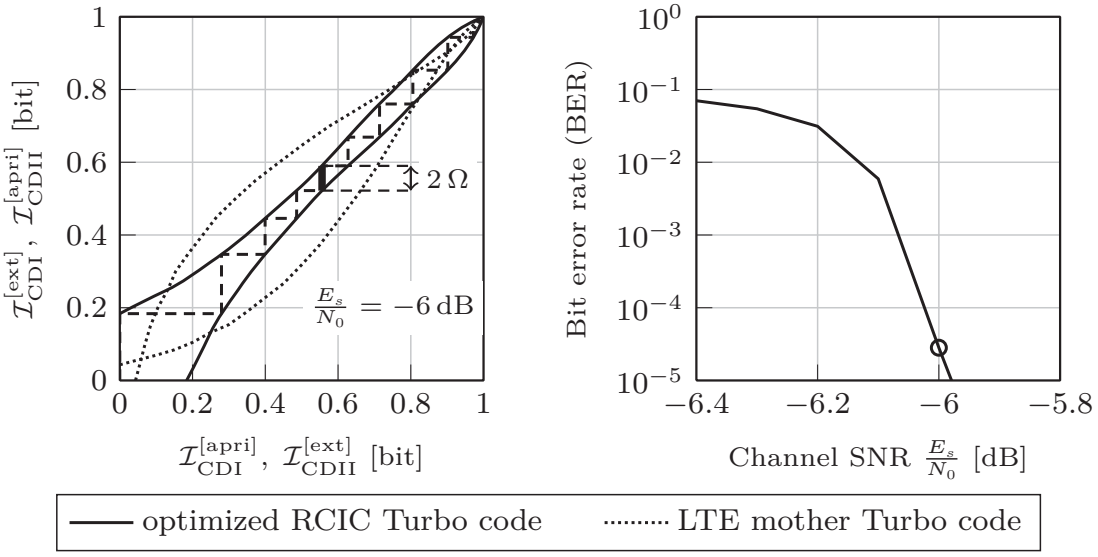


Figure 4.19: EXIT chart optimization at $E_s/N_0 = -6\,\mathrm{dB}$ for the systematic rate-1/3 LTE Turbo code. Two EXIT charts are plotted (left hand side): EXIT chart for the mother code (dotted lines) and the EXIT chart of the semi-analytically optimized RCIC Turbo code (bold lines) including its decoding trajectory (dashed line). Furthermore, the corresponding BER is shown (right hand side).

4.3.3 Deterministic Insertion

In Figure 4.20, the derived EXIT functions for the system with randomly inserted dummy bits (dashed lines) are compared to the measured EXIT functions of a system employing the deterministic insertion procedure described in Algorithm 3.1 (dashed lines). Both RCIC codes are based on the systematic rate-1/2 LTE component code with generator polynomials $\mathbf{G}_{\mathrm{CC}} = (1, 15/13)_8$ and constraint length $\mathfrak{L} + 1 = 4$. The evaluation has been carried out for dummy bit fractions $P_{\mathrm{D}} = \ell_d/\ell_x \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$ resulting in code rates after dummy bit insertion of $r_{\mathrm{IC}} \in \{0.31, 0.29, 0.26, 0.23, 0.20, 0.17, 0.13\}$ according to (3.5). Transmission has been performed over a BEC with erasure probability $P_{\mathrm{C}} = 0.7$ as well as a BIAWGNC with channel SNR $E_s/N_0 = -6\,\mathrm{dB}$. It has been observed that slightly superior performance can be expected for the deterministic insertion in particular for large dummy bit fractions. Nevertheless, both EXIT functions match almost perfectly for dummy bit fractions of up to 40 %. Consequently, the derived expressions for RCIC codes with random insertion are well suited to predict the behavior of coding systems employing RCIC codes based on the proposed deterministic insertion rule.

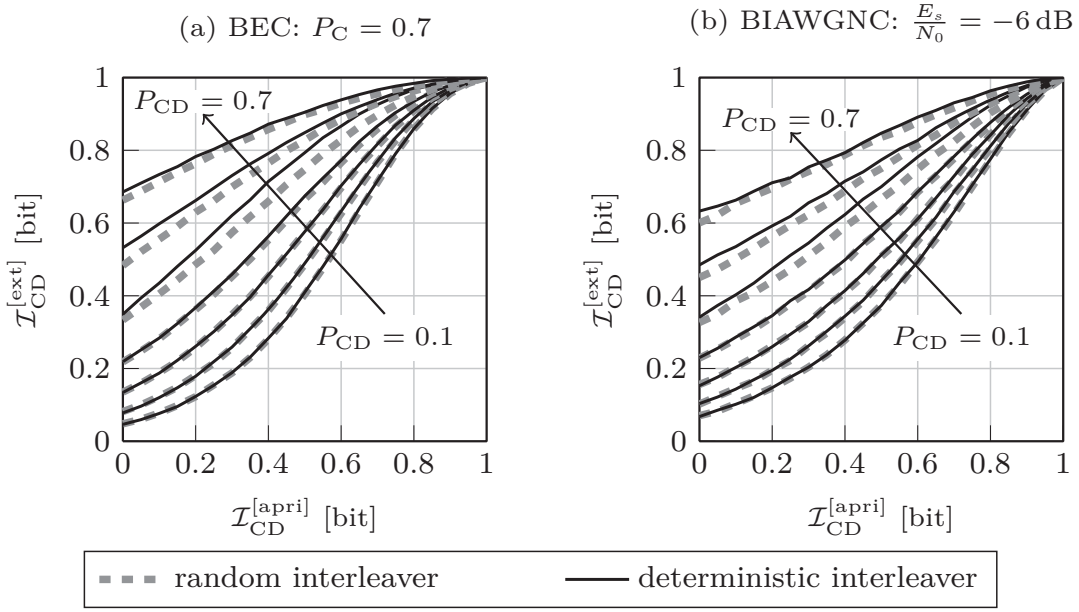


Figure 4.20: RCIC code exploiting deterministically inserted dummy bits according to Algorithm 3.1 (solid lines) versus RCIC code exploiting randomly inserted dummy bits (dashed lines). Convolutional mother code: $\mathbf{G}_{\mathrm{CC}} = (1, 15/13)_8$ (LTE).

4.4 EXIT Chart Comparison of RCIC and RCRC Turbo Codes

The expressions given in Sections 4.2 and 4.3 provide an excellent basis for the comparison of RCIC and RCRC Turbo codes in terms of decoding performance. It is worth noting that the following evaluation is, in principle, only valid for RCIC Turbo codes with random dummy bit insertion and RCRC Turbo codes with random bit repetition. However, it provides a useful approximation of more sophisticated insertion/repetition schemes as it has exemplarily been shown for the deterministic LTE bit repetition scheme in Section 4.2.3 and equidistantly inserted dummy bits in Section 4.3.3.

Performance evaluation is exemplarily carried out for the systematic rate-1/3 LTE Turbo code based on the information theoretic transmission models presented in Sections 4.2.1 and 4.3.1. Transmission is performed over a BEC or a BIAWGNC. For comparison reasons, it is assumed that near error-free decoding can be realized if at least an infinitesimal decoding tunnel $\Omega \to 0$ exists within the EXIT chart.

The objective of rate matching is to maximize the overall code rate subject to a fixed channel quality for which near error-free decoding is desired. Accordingly, the amount of redundancy required for each channel quality is determined individually for both methods. With the optimization constraints given in Sections 4.2.2 and 4.3.2, this can be carried out semi-analytically, which reduces considerably the number of required EXIT chart simulations.

The overall code rates r_{IC} and r_{RC} of the RCIC and RCRC Turbo codes are then given according to

$$r_{\mathrm{IC}} = \frac{1 - P_{\mathrm{D}}}{r_{\mathrm{PCCC}}^{-1} - P_{\mathrm{D}}} \quad , \quad r_{\mathrm{RC}} = r_{\mathrm{PCCC}} \cdot r_{\mathrm{R}}, \tag{4.62}$$

with P_{D} denoting the fraction of inserted dummy bits, r_{R} the fraction of repeated code bits, and r_{PCCC} the code rate of the employed mother Turbo code. Both code rates are depicted in Figure 4.21 as a function of P_{C} and E_s/N_0, respectively. The curves corresponding to the RCIC Turbo codes are plotted in bold lines, while the curves of the RCRC Turbo codes are marked by dashed lines.

For the considered LTE Turbo code, dummy bit insertion shows an improved effectiveness in nearly all cases compared to bit repetition, since a higher code rate is sufficient for near error-free decoding. Only for the transmission over a BIAWGNC there exists an intersection at approximately $-8.6\,\mathrm{dB}$ which corresponds to an effective code rate after rate matching of $r_{\mathrm{RM}} = r_{\mathrm{RC}} = r_{\mathrm{IC}} \approx 0.14$. Consequently, rate matching by dummy bit insertion should be employed for target code rates higher than approximately 1/7. This corresponds to an insertion of 67 % dummy bits before Turbo encoding. However, depending on the channel quality, gains of up to $\Delta P_{\mathrm{C}} = 0.02$ and $\Delta\mathrm{SNR} = 0.5\,\mathrm{dB}$, respectively, can be achieved for an equal target code rate. It is evident that the conclusion which type of rate

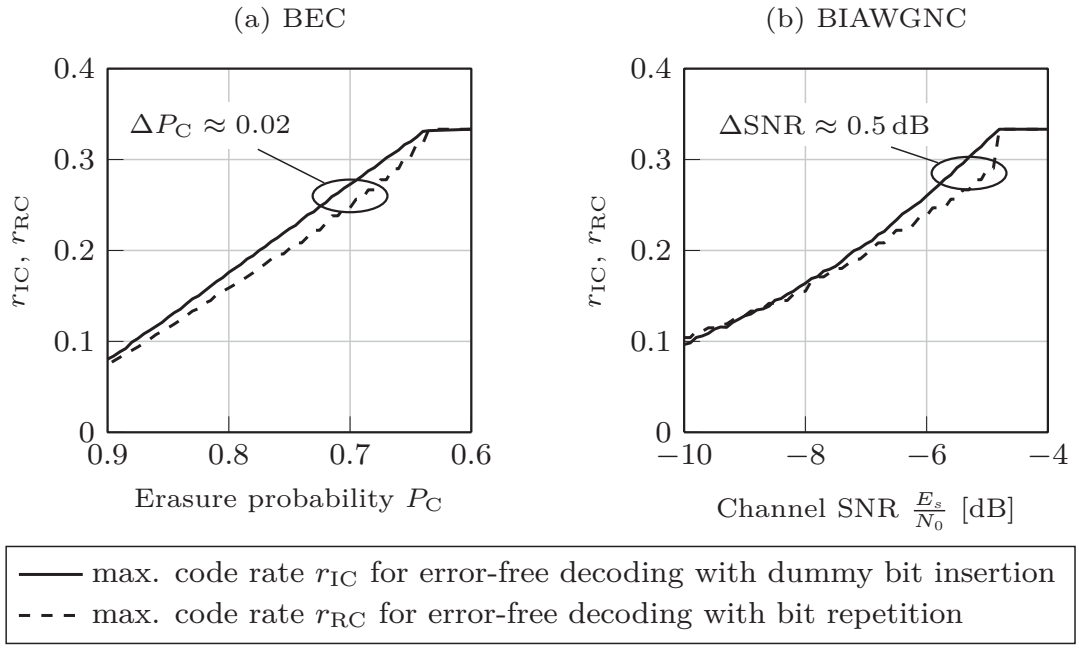


Figure 4.21: Comparison of RCRC and RCIC Turbo codes by means of EXIT charts. The LTE Turbo code is employed as mother code. Communication channel: BEC & BIAWGNC.

matching is to be preferred and which gains can be expected depends on the considered mother Turbo code and cannot be answered in general. However, for a widely used representative, rate matching by dummy bit insertion shows slightly superior performance and should be employed.

4.5 Generalization of EXIT Chart Analysis to BSC and LDPC Codes

In the previous sections, the influence of bit puncturing, bit repetition, and dummy bit insertion on the system performance has been analyzed using their EXIT charts. It has been shown by means of information theory how the EXIT charts of the underlying convolutional mother code have to be adapted accordingly. However, the expressions that have been derived in the previous sections are only valid for communication channels and *extrinsic* channels modeled as BECs or BIAWGNCs. Therefore, Section 4.5.1 aims at extending this analysis to another common channel model, namely the *Binary Symmetric Channel* (BSC).

Furthermore, it will be sketched in Section 4.5.2 how the proposed EXIT chart analysis of rate-compatible convolutional codes can be adapted to rate-compatible LDPC codes, extending the area of application.

4.5.1 Binary Symmetric Channel

All rate matching schemes analyzed throughout this chapter have been modeled by a transmission over so-called *equivalent* channels which are, in general, a serial or parallel concatenation of appropriate channel models. The core issue has been the derivation of the mutual information at the output of the *equivalent* channels. So far, these derivations have been carried out for a transmission over a BEC or a BIAWGNC, however, they will be adapted to the scenario in which the communication channel can be modeled as a BSC in what follows.

In the case of bit puncturing, a serial concatenation of a BEC with erasure probability P_{E}, modeling the puncturing effect, and a BSC with crossover (bit error) probability P_0, modeling the communication channel, is considered. The corresponding channel model is depicted in Figure 4.22 and is described by the mapping chain $\mathfrak{U} \mapsto \mathfrak{V} \mapsto \mathfrak{W} : \ \mathbb{F}_2 = \{0,1\} \to \mathbb{V} = \{0,1,*\} \to \mathbb{W} = \{0,1,*\}$, where $\mathfrak{U}$, $\mathfrak{V}$ and $\mathfrak{W}$ signify random variables and $\mathfrak{V} = \mathfrak{W} = *$ a bit erasure at the output of the channel (see Figure 4.22(a)). The *equivalent* channel is shown in Figure 4.22(b) and is defined by the transition probabilities P_{E}, $P_1 = (1-P_{\mathrm{E}}) \cdot P_0$, and $P_2 = (1-P_{\mathrm{E}}) \cdot (1-P_0)$. Hence, it remains to determine the mutual information $\mathcal{I}^{\mathrm{ser}}(\mathfrak{W};\mathfrak{U})$ at the output of the *equivalent* channel.

Proposition 4.5.1 *Consider a BEC with erasure probability P_{E} and mutual in-*

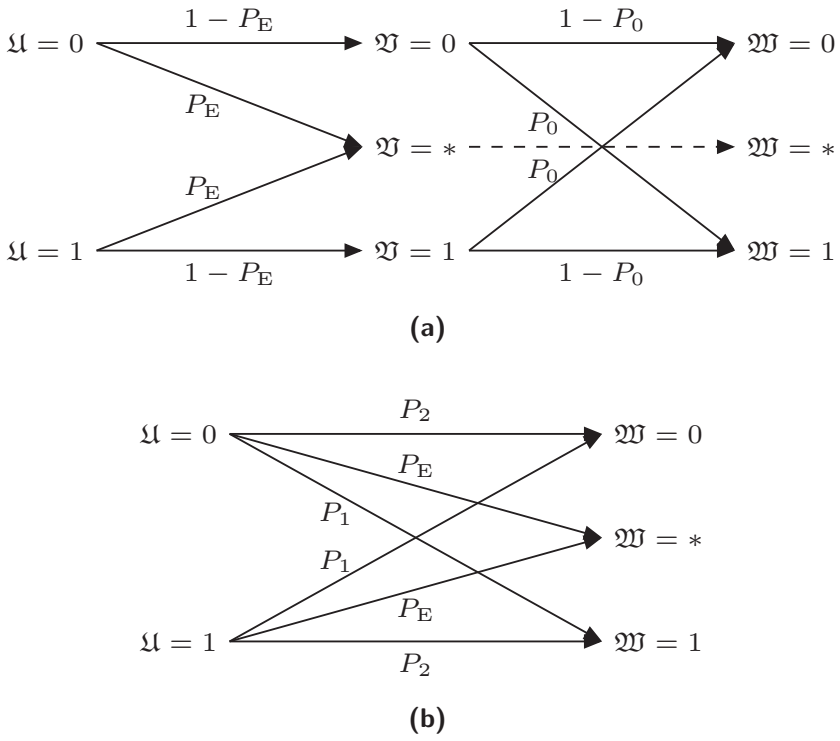


Figure 4.22: (a) Serial concatenation of a BEC and a BSC. (b) *Equivalent* channel model with $P_1 = (1-P_{\mathrm{E}}) \cdot P_0$, and $P_2 = (1-P_{\mathrm{E}}) \cdot (1-P_0)$.

formation $\mathcal{I}_{\mathrm{BEC}}(\mathfrak{V};\mathfrak{U}) = 1 - P_{\mathrm{E}}$ *which is serially concatenated to a BSC with crossover probability* P_0 *and mutual information* $\mathcal{I}_{\mathrm{BSC}}(\mathfrak{W};\mathfrak{V}) = 1 - H'(P_0) = 1 - (-P_0 \cdot \log_2(P_0) - (1-P_0) \cdot \log_2(1-P_0))$. *The function* $H'(\cdot)$ *signifies the binary entropy function for uniformly distributed input bits. Then, the mutual information* $\mathcal{I}^{\mathrm{ser}}(\mathfrak{W};\mathfrak{U})$ *at the output of the serial concatenation is given by*

$$\begin{aligned}\mathcal{I}^{\mathrm{ser}}(\mathfrak{W};\mathfrak{U}) &= \mathcal{I}_{\mathrm{BEC}}(\mathfrak{V};\mathfrak{U}) \cdot \mathcal{I}_{\mathrm{BSC}}(\mathfrak{W};\mathfrak{V}) \\ &= (1 - P_{\mathrm{E}}) \cdot (1 - H'(P_0)) \\ &= (1 - P_{\mathrm{E}}) \cdot (1 + P_0 \cdot \log_2(P_0) + (1-P_0) \cdot \log_2(1-P_0)) \end{aligned} \tag{4.63}$$

Proof 4.5.1 *The proof is given in Appendix B.*

As described in Section 4.2, bit repetition with arbitrary repetition rate r_{R} can be modeled by a parallel concatenation of Q channels, each consisting of a serial concatenation of a BEC with a specific communication channel. Modeling the communication channel as a BSC, the *equivalent* communication channel depicted in Figure 4.8 consists of a parallel concatenation of the channel shown in Figure 4.22. It is described by the mapping $\mathfrak{U} \mapsto \mathfrak{W} = (\mathfrak{W}_1, \ldots, \mathfrak{W}_q, \ldots, \mathfrak{W}_Q) : \mathbb{F}_2 \to \mathbb{W}^Q$ with $\mathfrak{w} = (\mathfrak{w}_1, \ldots, \mathfrak{w}_q, \ldots, \mathfrak{w}_Q)$ signifying a realization of $\mathfrak{W}$. The mutual information at the output of that channel can be computed according to Proposition 4.5.2.

Proposition 4.5.2 *Consider* Q *parallel concatenated channels, each with mutual information* $\mathcal{I}_q^{\mathrm{ser}}(\mathfrak{W}_q;\mathfrak{U}) = (1 - P_{\mathrm{E},q}) \cdot (1 - H'(P_0))$, $(1 \le q \le Q)$, *given according to Proposition 4.5.1. Then, the mutual information* $\mathcal{I}^{\mathrm{par},1}(\mathfrak{W};\mathfrak{U})$ *at the output of the parallel concatenation is given by*

$$\mathcal{I}^{\mathrm{par},1}(\mathfrak{W};\mathfrak{U}) = -\sum_{\mathfrak{w} \in \mathbb{W}^Q} P_{\mathfrak{W}}(\mathfrak{w}) \log_2(P_{\mathfrak{W}}(\mathfrak{w})) - \sum_{q=1}^{Q} H(\mathfrak{W}_q|\mathfrak{U}) \tag{4.64}$$

with

$$P_{\mathfrak{W}}(\mathfrak{w}) = \frac{1}{2} \cdot \sum_{u \in \mathbb{F}_2} \prod_{q=1}^{Q} P_{\mathfrak{W}_q|\mathfrak{U}}(\mathfrak{w}_q|\mathfrak{u}), \tag{4.65}$$

$$P_{\mathfrak{W}_q|\mathfrak{U}}(\mathfrak{w}_q|\mathfrak{u}) = \begin{cases} (1 - P_{\mathrm{E},q}) \cdot (1 - P_0) & : \quad \mathfrak{w}_q = \mathfrak{u} \\ (1 - P_{\mathrm{E},q}) \cdot P_0 & : \quad \mathfrak{w}_q = 1 - \mathfrak{u} \\ P_{\mathrm{E},q} & : \quad \mathfrak{w}_q = * \end{cases}, \quad \mathfrak{u} \in \mathbb{F}_2 \tag{4.66}$$

and

$$\begin{aligned} H(\mathfrak{W}_q|\mathfrak{U}) = &-P_{\mathrm{E},q} \cdot \log_2(P_{\mathrm{E},q}) - (1 - P_{\mathrm{E},q}) \cdot P_0 \cdot \log_2((1 - P_{\mathrm{E},q}) \cdot P_0) \\ &- (1 - P_{\mathrm{E},q}) \cdot (1 - P_0) \cdot \log_2((1 - P_{\mathrm{E},q}) \cdot (1 - P_0)). \end{aligned} \tag{4.67}$$

Proof 4.5.2 *The proof is given in Appendix B and is strongly related to the proof given in [LH06] for parallel concatenated BSCs.*

Furthermore, an EXIT chart analysis of RCIC (Turbo) codes was introduced in Section 4.3. The corresponding information theoretic transmission model (see Figure 4.15), in which the communication channel and the *extrinsic* channel are now modeled as BSCs, comprises a parallel concatenated BEC and BSC. As shown in Appendix B (proof of Proposition 4.3.1), (4.56) is valid for arbitrary *extrinsic* channels and, thus, also for a BSC:

$$\begin{aligned}\mathcal{I}^{\mathrm{par},2} &= \mathcal{I}_{\mathrm{BSC}} + \mathcal{I}_{\mathrm{BEC}} \cdot (1 - \mathcal{I}_{\mathrm{BSC}}) \\ &= 1 - H'(P_0) \cdot P_{\mathrm{E}}. \end{aligned} \tag{4.68}$$

Consequently, the EXIT functions for RCPC, RCRC, and RCIC codes for the transmission over a BSC can be determined based on the EXIT function of the underlying convolutional mother code by substituting the expressions (4.1), (4.21), and (4.56) with (4.63), (4.64), and (4.68), respectively. The crossover probability of the *equivalent* communication channel in the RCPC and RCRC transmission model is given by $\mathring{P}_0 = (H')^{-1}(\mathcal{I}^{\mathrm{ser}})$ and $P_0^{\Sigma} = (H')^{-1}(\mathcal{I}^{\mathrm{par},1})$, respectively. The binary entropy function $H'(\cdot)$ is strictly monotonically increasing within $0 \leq P_0 \leq 0.5$ and, thus, invertible. However, there does not exist an analytical solution, i.e., the inverse function has to be approximated numerically.

For validation, EXIT functions of the LTE component code with generator polynomial(s) $\mathbf{G}_{\mathrm{CC}} = (1, 15/13)_8$, constraint length $\mathfrak{L} + 1 = 4$, and code rates after convolutional coding and rate matching of $r_{\mathrm{RM}} \in \{1/6, 1/5, 1/4, 1/3, 2/3, 3/4, 4/5\}$ are shown in Figure 4.23. Code rates higher than 1/2 are generated by means of random bit puncturing (Figure 4.23(a)) and code rates smaller than 1/2 are provided using bit repetition (Figure 4.23(b)). In case (a), transmission is carried out over a BSC with crossover probability $P_0 = 0.1$, while in case (b) the crossover probability is set to $P_0 = 0.2$. Both figures reveal that there is a deviation between the measured and the derived EXIT functions. This deviation is caused by the following reason. In order to adapt the EXIT function of the underlying mother code in accordance to the influence of the respective rate matching scheme, it has to be assumed that the *equivalent* communication channel can be modeled by the same channel model as the communication channel (see Figures 4.1 and 4.7). This assumption warranties that the mutual information at the output of the *equivalent* communication channel can directly be transformed into an *equivalent* crossover probability $\mathring{P}_0$ or P_0^{Σ}, respectively. However, this assumption is only valid for transmission over a BEC as already mentioned in Section 4.1. For the BSC case, the channel model of the *equivalent* channel is given according to Figure 4.22 (RCPC codes) or a parallel concatenation of that channel (RCRC codes). Consequently, the channel model assumption is violated yielding minor differences in the probability distribution of the output LLRs and, thus, in the derived EXIT functions

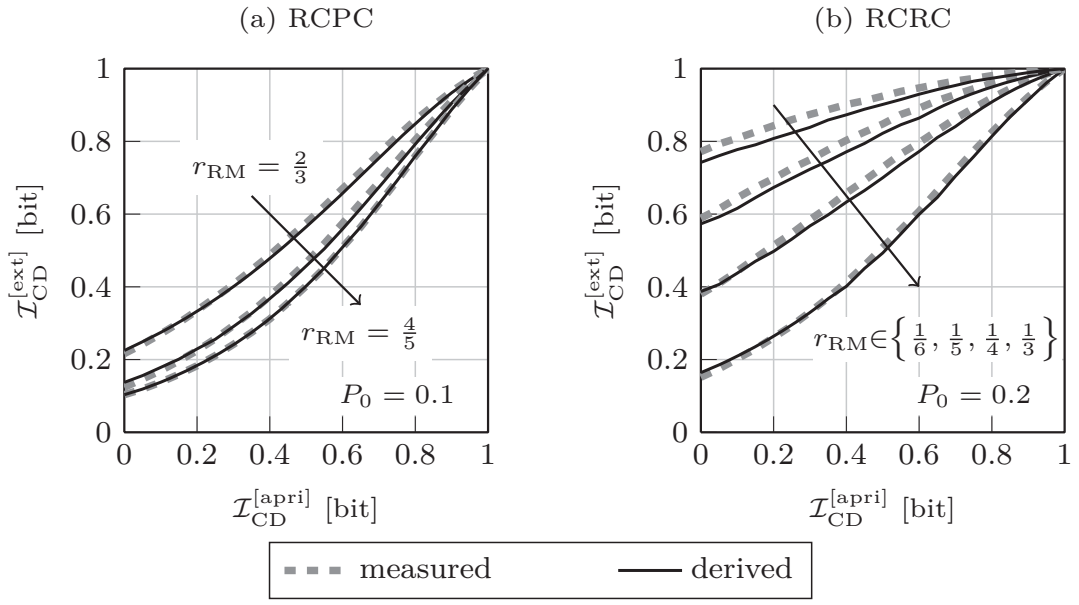


Figure 4.23: Comparison of derived and measured EXIT functions for a transmission over a BSC. EXIT functions for RCPC codes (left) and RCRC codes (right) are depicted.

which are not identical to their measured versions. Nevertheless, they are close approximations in most of the considered cases.

4.5.2 Low-Density Parity-Check Codes

LDPC codes are commonly decoded using message passing algorithms as, e.g., belief propagation [Pea88]. *Extrinsic* information $\mathbf{L}_{\mathrm{VN}}^{[\mathrm{ext}]}(\mathfrak{e})$ and $\mathbf{L}_{\mathrm{CN}}^{[\mathrm{ext}]}(\mathfrak{e})$ on $\mathfrak{e} = (\mathfrak{e}_1, \ldots, \mathfrak{e}_i, \ldots, \mathfrak{e}_{\ell_\mathfrak{e}}) \in \mathbb{R}^{\ell_\mathfrak{e}}$ is exchanged between the *Variable Nodes* (VN) and the *Check Nodes* (CN) along the edges of the bipartite Tanner graph [Tan81] as illustrated in Figure 4.24. Each $\mathfrak{e}_i$ is associated with one edge within the Tanner graph and corresponds to the value of the variable node to which the respective edge is connected. Using the notation introduced in Section 2.2.2, each edge signifies a connection between one distinct check node ν ($1 \leq \nu \leq \ell_c$) and one distinct variable node n ($1 \leq n \leq \ell_y$). Thus, the *extrinsic* information $L_{\mathrm{VN}}^{[\mathrm{ext}]}(\mathfrak{e}_i)$ and $L_{\mathrm{CN}}^{[\mathrm{ext}]}(\mathfrak{e}_i)$ correspond to a certain variable node message $\mathfrak{v}_{\nu,n}$ and a certain check node message $\mathfrak{c}_{\nu,n}$, respectively, which is calculated by means of (2.18) and (2.16). After each iteration, the information bits $\hat{\mathbf{b}} = (\hat{y}_1, \ldots, \hat{y}_n, \ldots, \hat{y}_{\ell_b})$ are estimated based on the first $\ell_b < \ell_y$ *a posteriori* LLRs $\mathbf{L}_{\mathrm{VN}}(\mathbf{b}) = \big(L_{\mathrm{VN}}(y_1), \ldots, L_{\mathrm{VN}}(y_n), \ldots, L_{\mathrm{VN}}(y_{\ell_b})\big)$ according to (2.19). The iteration process is stopped if all ℓ_c check equations defined by the parity-check matrix $\mathbf{A}$ are fulfilled or the maximum number of iterations is reached.

A general transmission model for the EXIT chart analysis which is suitable for LDPC codes was introduced by Ashikhmin et al. in [AKtB02, AKtB04]. This model is capable of analyzing both, the behavior of the VN decoder as well as

the behavior of the CN decoder. For the sake of clarity, the analysis of both component decoders is carried out separately based on the models depicted in Figure 4.25. Considering Figure 4.25(a) at first, an information bit vector $\mathbf{b}$ of length ℓ_b is encoded by an LDPC code, which appends $\ell_c = \ell_y - \ell_b$ parity bits to the input vector. The output vector $\mathbf{y}$ is then transmitted over the communication channel with mutual information $\mathcal{I}_\mathrm{C}$ delivering channel-related LLRs $\mathbf{L}_\mathrm{DM}^{[\mathrm{chan}]}(\mathbf{y})$ to the SISO VN decoder. Furthermore, *a priori* information $\mathbf{L}_\mathrm{VN}^{[\mathrm{apri}]}(\mathfrak{e}) = \mathbf{L}_\mathrm{CN}^{[\mathrm{ext}]}(\mathfrak{e})$ generated by the CN decoder is provided to the VN decoder. This is modeled by a serial concatenation of a VN→Edge mapper and an *extrinsic* channel with mutual information $\mathcal{I}_\mathrm{VN}^{[\mathrm{apri}]}$. The role of the mapper is to map the code bits onto the edges of the Tanner graph according to the parity-check matrix $\mathbf{A}$. The *extrinsic* channel subsequently models the reliability values of the messages transmitted from the check nodes to the variable nodes. The channel-related information $\mathbf{L}_\mathrm{DM}^{[\mathrm{chan}]}(\mathbf{y})$ as well as the *a priori* information $\mathbf{L}_\mathrm{VN}^{[\mathrm{apri}]}(\mathfrak{e})$ are then exploited by the SISO VN decoder to compute the *extrinsic* information $\mathbf{L}_\mathrm{VN}^{[\mathrm{ext}]}(\mathfrak{e})$ and the *a posteriori* information $\mathbf{L}_\mathrm{VN}(\mathbf{b})$.

The EXIT model for the check node decoder, which is shown in Figure 4.25(b), equals the lower branch in Figure 4.25(a), since no channel-related information is given to the SISO CN decoder. Only *a priori* information $\mathbf{L}_\mathrm{CN}^{[\mathrm{apri}]}(\mathfrak{e}) = \mathbf{L}_\mathrm{VN}^{[\mathrm{ext}]}(\mathfrak{e})$ after VN→Edge mapping and transmission over an *extrinsic* channel with mutual information $\mathcal{I}_\mathrm{CN}^{[\mathrm{apri}]}$ can be exploited by the SISO CN decoder to determine the *extrinsic* information $\mathbf{L}_\mathrm{CN}^{[\mathrm{ext}]}(\mathfrak{e})$.

In the previous sections, the influence of various rate matching schemes on the EXIT charts of convolutional Turbo codes has been evaluated by means of information theory. However, such an analysis is not limited to Turbo codes and can be performed for LDPC codes as well. Bit puncturing and bit repetition, e.g., can be modeled by an identical *equivalent* communication channel (see Figure 4.1 and Figure 4.8, respectively) with either decreased or increased mutual information. Therefore, the expressions derived for Turbo codes with random bit puncturing or

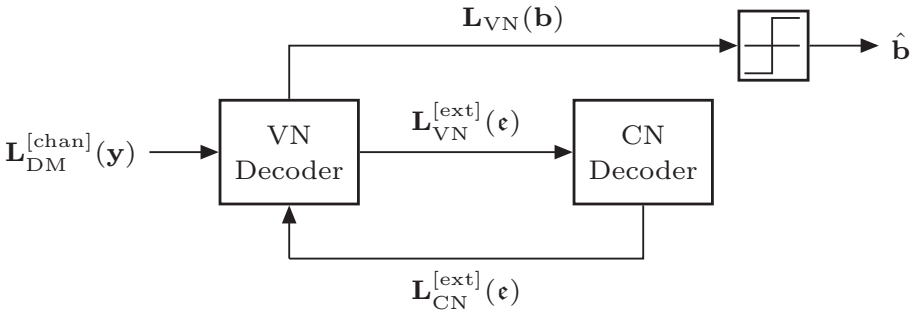
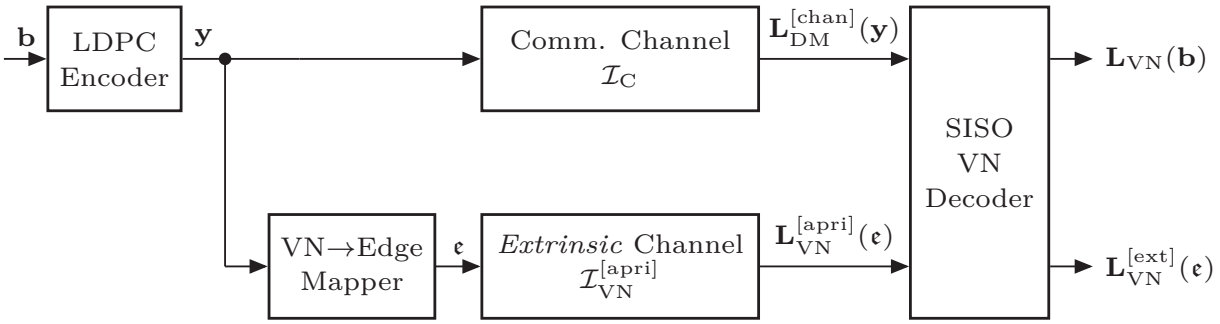


Figure 4.24: Block diagram of an LDPC decoder. *Extrinsic* information is iteratively exchanged between a *Variable Node* (VN) decoder and a *Check Node* (CN) decoder.



(a) Information theoretic transmission model for the EXIT chart analysis of the variable node decoder.

(b) Information theoretic transmission model for the EXIT chart analysis of the check node decoder.

Figure 4.25: Models suitable for the EXIT chart analysis of LDPC codes.

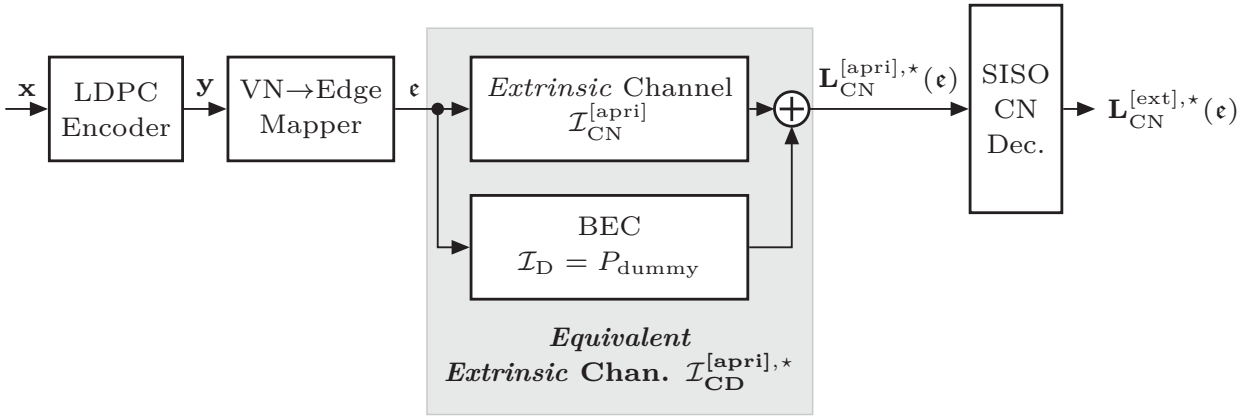


Figure 4.26: EXIT model for the check node decoder if rate matching by dummy bit insertion is employed prior to LDPC encoding.

random bit repetition can also be used for the determination of the variable node EXIT function. In the case of rate matching by dummy bit insertion, the expressions derived for RCIC codes have to be slightly adapted following the modified EXIT model for the check node decoder, which is depicted in Figure 4.26. The input vector $\mathbf{x}$ of length ℓ_x contains ℓ_d dummy bits and ℓ_b information bits. Furthermore, a BEC with mutual information $\mathcal{I}_{\mathrm{D}} = P_{\mathrm{dummy}}$ is connected in parallel to the *extrinsic* channel. This BEC models the additional reliability information delivered to the SISO CN decoder due to dummy bit insertion, where P_{dummy} denotes the fraction of known messages $\mathfrak{e}$. The parallel concatenation is substituted

by an *equivalent extrinsic* channel with increased mutual information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],\star}$. In contrast to the RCIC case, P_{dummy} is determined by the fraction $P_{\mathrm{D}} = \ell_d/\ell_x$ of dummy bits *and* the fraction P_{IN} of edges connected to *Information Nodes* (IN), i.e. nodes which correspond to information bits:

$$P_{\mathrm{dummy}} = P_{\mathrm{D}} \cdot P_{\mathrm{IN}} = \frac{\ell_d}{\ell_x} \cdot \frac{\sum_{n=1}^{\ell_b} \sum_{\nu=1}^{\ell_c} A_{\nu,n}}{\sum_{n=1}^{\ell_y} \sum_{\nu=1}^{\ell_c} A_{\nu,n}}. \tag{4.69}$$

In (4.69), each value $A_{\nu,n}$ corresponds to an entry in the parity-check matrix $\mathbf{A}$. Substituting P_{D} with P_{dummy} and $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}$ with $\mathcal{I}_{\mathrm{CN}}^{[\mathrm{apri}]}$ in (4.56) results in the desired mutual information

$$\mathcal{I}_{\mathrm{CN}}^{[\mathrm{apri}],\star} = \mathcal{I}_{\mathrm{CN}}^{[\mathrm{apri}]} + P_{\mathrm{dummy}} \cdot \left(1 - \mathcal{I}_{\mathrm{CN}}^{[\mathrm{apri}]}\right) \tag{4.70}$$

and, finally, in the desired expression for the EXIT function $\mathcal{T}_{\mathrm{CN}}^{\star}$ after dummy bit insertion:

$$\mathcal{T}_{\mathrm{CN}}^{\star}\left(\mathcal{I}_{\mathrm{CN}}^{[\mathrm{apri}]}\right) = \mathcal{T}_{\mathrm{CN}}\left(\mathcal{I}_{\mathrm{CN}}^{[\mathrm{apri}],\star}\right). \tag{4.71}$$

The EXIT function of the mother LDPC code is denoted as $\mathcal{T}_{\mathrm{CN}}$ in this case.

As an example, the rate-1/3 irregular LDPC code with degree distributions (node perspective) $\lambda(\chi) = 0.565771 \cdot \chi^2 + 0.211528 \cdot \chi^3 + 0.118725 \cdot \chi^4 + 0.103976 \cdot \chi^{11}$ and $\varrho(\chi) = 0.922857 \cdot \chi^5 + 0.0771428 \cdot \chi^6$ is employed as mother code of the considered rate-compatible LDPC code. The degree distributions have been optimized with density evolution [RSU01] and the *Progressive Edge Growth* (PEG) algorithm proposed by Hu et al. [HEA05] has been used to generate the LDPC code. Code rate adaptation is carried out by dummy bit insertion. A fraction of $P_{\mathrm{D}} \in \{0.2, 0.4, 0.6, 0.8\}$ dummy bits is randomly inserted into the information bit vector before LDPC encoding. For the considered mother code with $P_{\mathrm{IN}} = 0.423$, this results in a fraction of known (dummy) edges $P_{\mathrm{dummy}} \in \{0.086, 0.172, 0.258, 0.344\}$. The communication channel is either modeled as a BEC with erasure probability $P_{\mathrm{C}} = 0.6$ or a BIAWGNC with signal-to-noise ratio $E_s/N_0 = -4\,\mathrm{dB}$. The *extrinsic* channel is likewise modeled as BEC or BIAWGNC. The corresponding EXIT functions $\mathcal{T}_{\mathrm{CN}}^{\star}$ of the check node decoder are shown in Figure 4.27. The EXIT functions of the variable node decoder are omitted in these figures, since dummy bit insertion has only an influence on the check node decoder performance. Furthermore, all EXIT functions are plotted with swapped axes according to the common EXIT chart representation. While in the BEC case, the derived EXIT functions match perfectly their measured versions, minor deviations can be observed in the BIAWGNC case, since the LLRs $\mathbf{L}_{\mathrm{CN}}^{[\mathrm{apri}],\star}(\mathfrak{e})$ at the output of the parallel concatenated channel are not Gaussian distributed. However, the information theoretical

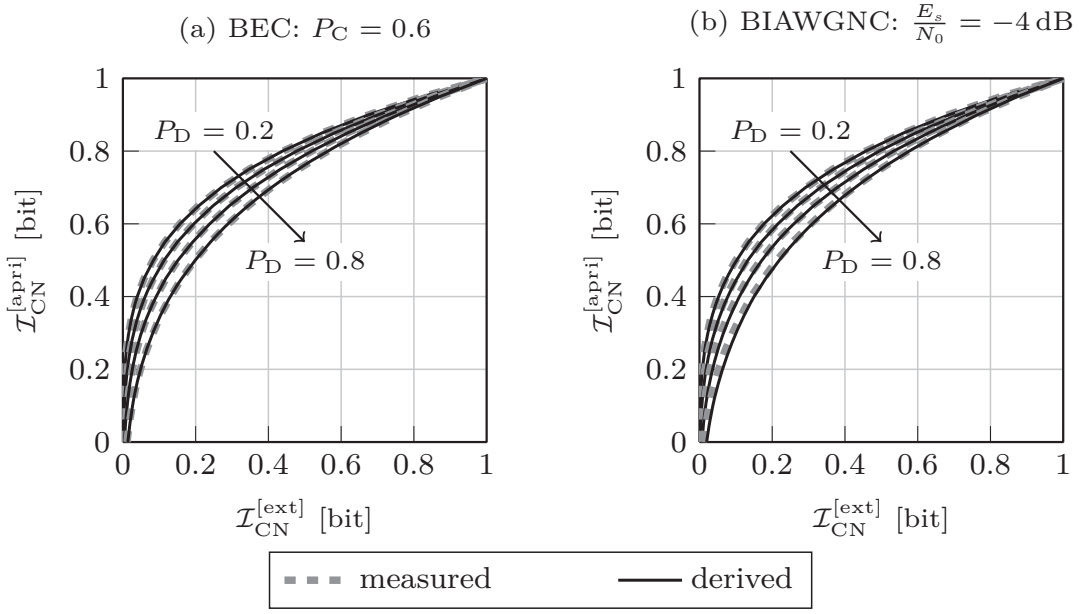


Figure 4.27: Comparison of derived (bold lines) and measured EXIT functions (dashed lines) for the check node decoder of a rate-compatible LDPC code. Dummy bit insertion with different dummy bit fractions $P_D \in \{0.2, 0.4, 0.6, 0.8\}$ is employed for code rate adaptation. Transmission is simulated over a BEC (a) or a BIAWGNC (b).

modeling allows for an approximation of the EXIT functions after dummy bit insertion based on the EXIT function of the mother code with high accuracy.

It is worth noting that the EXIT chart analysis based on such a model is, in principle, inaccurate for the BIAWGNC case since both input LLR vectors $\mathbf{L}_{DM}^{[chan]}(\mathbf{y})$ and $\mathbf{L}_{CN}^{[apri]}(\mathfrak{e})$ are modeled by Gaussian random variables. While the channel-related LLRs are Gaussian distributed, the *extrinsic* LLRs severely differ from the assumed Gaussian distribution due to the box-plus operation carried out by the check nodes. Therefore, EXIT charts determined by Gaussian approximation might not reflect the actual convergence behavior of the system. An alternative method to construct EXIT charts for LDPC codes is based on the simulation of the decoding trajectory. The drawback of this method is that the number of simulated points depends on the number of required iterations for near error-free decoding. If a low number of iterations is sufficient, the EXIT chart accuracy suffers from the low number of supporting points. Nevertheless, both methods are conventionally employed in literature.

Chapter 5

Rate Matching by Dummy Bit Insertion

Rate matching is used in many modern mobile and wireless communication systems to adapt the error protection to the instantaneous channel condition. The design differs from system to system and depends on the employed channel coding concept. As described in Section 3.4, LTE uses bit puncturing and bit repetition in order to flexibly adapt the code rate of the fixed-rate convolutional mother Turbo code. This differs from WiMAX (Worldwide Interoperability for Microwave Access), in which an ensemble of different *Low-Density Parity-Check* (LDPC) codes with individual code rates and input block sizes are defined to perform rate matching. It is evident that such a realization suffers from a low code rate flexibility since one encoder/decoder pair has to be specified for each distinct target code rate.

Rate matching by dummy bit insertion is an effective alternative to those rate matching schemes. In LTE, e.g., it can be employed instead of bit repetition resulting in an improved decoding performance (Section 5.1) [BEV11]. In WiMAX, it can be used to design a rate-compatible system based on a single, high-rate LDPC mother code [BBV11]. Any lower rates can then be generated by dummy bit insertion which provides a higher flexibility in terms of code rate adaptation compared to the conventional scheme without a significant performance degradation (Section 5.2).

It is worth mentioning that the application of dummy bit insertion is not limited to the communication systems considered in this chapter. It is, e.g., also an expedient alternative for systems based on *Iterative Source-Channel Decoding* (ISCD) [Gör00, AVS01] in which *extrinsic* information is exchanged between a source decoder (outer component) and a channel decoder (inner component). Significant gains in terms of decoding complexity could be realized in [BV10b] without any loss in performance by using a *Rate-Compatible Insertion Convolutional* (RCIC) code instead of a pure convolutional code (common setup) as inner component.

5.1 LTE

Rate matching by dummy bit insertion provides a flexible way of decreasing the overall code rate after convolutional Turbo coding. It is, thus, an alternative to

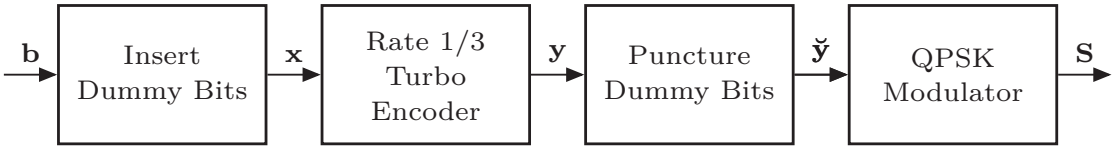


Figure 5.1: Rate matching by dummy bit insertion for LTE.

simple bit repetition as conventionally employed in LTE (see Section 3.4). According to the modified LTE transmitter model depicted in Figure 5.1, ℓ_d dummy bits are inserted into an information bit sequence $\mathbf{b}$ of length ℓ_b before rate-1/3 convolutional Turbo encoding resulting in a frame $\mathbf{x}$ of length $\ell_x = \ell_b + \ell_d$. All dummy bits appear within the output frame $\mathbf{y}$ of length $\ell_y = 3 \cdot (\ell_b + \ell_d)$ since a systematic Turbo code is employed in LTE. However, they are known in advance and do not have to be transmitted over the channel. The resulting frame after dummy bit puncturing $\tilde{\mathbf{y}}$ of length $\ell_{\tilde{y}} = 3 \cdot (\ell_b + \ell_d) - \ell_d$ is finally mapped to modulation symbols $\mathbf{S}$ using QPSK and transmitted over the channel. In what follows, the modified LTE system employing dummy bit insertion is compared to the conventional system using bit repetition in terms of their *Frame Error Rate* (FER) performance, convergence speed and system throughput.

In Figure 5.2, the residual FER is depicted dependent on the channel *Signal-*

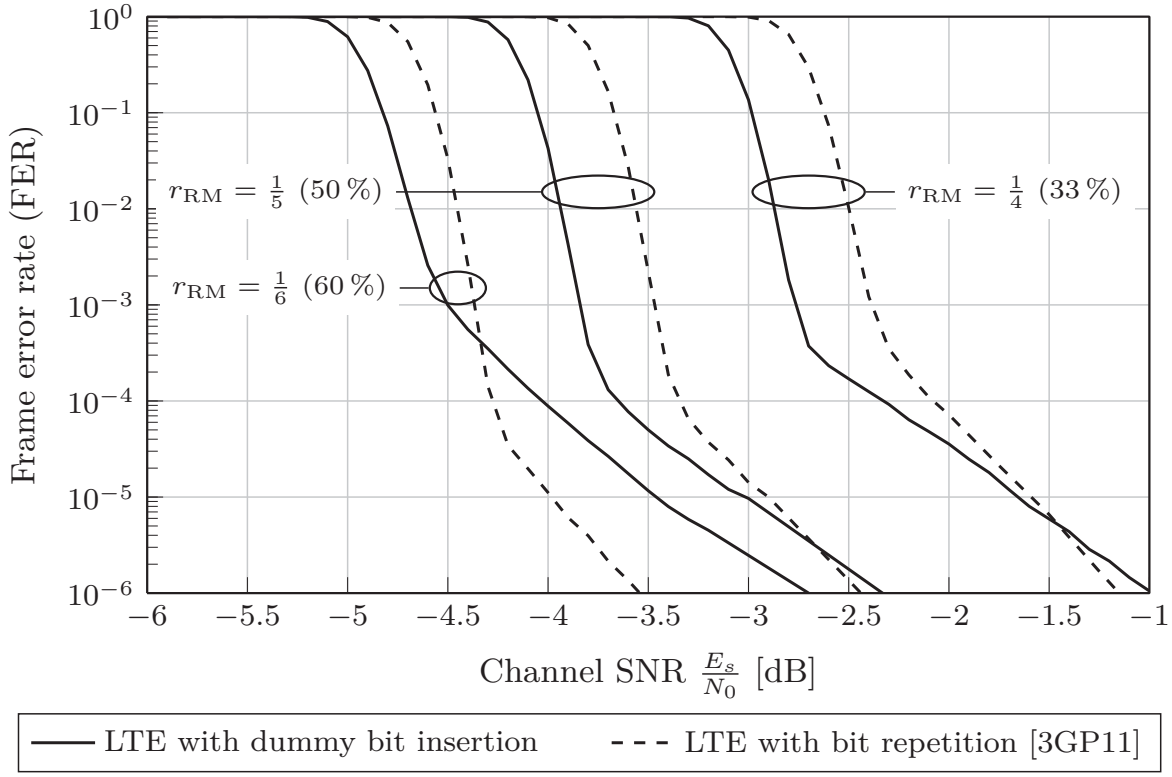


Figure 5.2: Frame error rate of the standardized LTE system for code rates $r_{\mathrm{RM}} \in \{1/4, 1/5, 1/6\}$ compared to the modified LTE system based on dummy bit insertion.

to-Noise Ratio (SNR) E_s/N_0 for target code rates $r_{\mathrm{RM}} \in \{1/4, 1/5, 1/6\}$ after rate-1/3 convolutional Turbo coding and rate matching either by means of bit repetition (dashed lines) or by dummy bit insertion (solid lines). Simulations have been conducted in an AWGN environment. The uncoded frame size was set to $\ell_b = 6144$ and 10 Turbo iterations were carried out at the receiver based on the LogMAP algorithm [RVH95, BCJR74]. Either bit repetition as specified in the LTE system [3GP11] or deterministic dummy bit insertion according to Algorithm 3.1 was employed in order to generate the desired target code rates. Taking (3.5) into account, the set of target code rates corresponds to dummy bit fraction of $\ell_d/\ell_x \in \{33\,\%, 50\,\%, 60\,\%\}$. Note that no *Hybrid Automatic Repeat reQuest* (HARQ) is considered in the following analysis. The influence of HARQ on the system performance will be analyzed in Section 6.1. In can be observed from Figure 5.2 that a gain in terms of E_s/N_0 of $\Delta\mathrm{SNR} \approx 0.4\,\mathrm{dB}$ is achieved in the waterfall region with the modified LTE system using dummy bit insertion for the target code rates $r_{\mathrm{RM}} = 1/4$ and $r_{\mathrm{RM}} = 1/5$. The error floor is comparable for both cases; the intersection of the respective curves is below 10^{-5}. However, for the target code rate $r_{\mathrm{RM}} = 1/6$, which corresponds to a coding system with 60 % inserted dummy bits, the gain is considerably decreased to $\Delta\mathrm{SNR} \approx 0.2\,\mathrm{dB}$. Furthermore, a higher error floor occurs for the modified system resulting in an intersection of both curves at an FER value of $4 \cdot 10^{-4}$ and at $E_s/N_0 = -4.4\,\mathrm{dB}$, respectively. The loss in performance is caused by the fact that more than 50 % dummy bits are inserted into the information bit sequence which entails a dummy bit distribution in which some dummy bits are directly adjacent. The information provided by these dummy bits is partly wasted reducing the gain in terms of FER due to the following reason. A convolutional (Turbo) code introduces strong dependencies between neighboring bits. The information delivered by the dummy bits can, thus, be used in particular to resolve the dependencies of directly adjacent bits rather than between more distant bits. Therefore, at least one information bit should be placed between two dummy bits which is only possible for dummy bit fractions less or equal than 50 %. This observation suggests that only target code rates of $1/5 \leq r_{\mathrm{RM}} \leq 1/3$ should be provided by dummy bit insertion. Lower rates can be constructed more efficiently, e.g., by a joint dummy bit insertion and bit repetition approach. However, note that small gains can also be achieved by pure dummy bit insertion for code rates lower than 1/5 with the drawback of a considerably higher error floor.

In Figure 5.3, the convergence behavior of both systems is analyzed. The residual FER is plotted in all four subfigures against the number of employed Turbo iterations. In (a) and (b), both systems are evaluated for a target code rate $r_{\mathrm{RM}} = 1/4$ (33 % dummy bits) and at two different channel SNRs $E_s/N_0 = -2.5\,\mathrm{dB}$ (waterfall region) and $E_s/N_0 = -2.1\,\mathrm{dB}$ (error floor region), respectively. Case (a): Assuming a target FER of 10^{-2}, 10 Turbo iterations are required for the conventional LTE system employing bit repetition while only 5 iterations are sufficient

for the modified system using dummy bit insertion. A similar observation can be made in case (b): In order to achieve a target FER of 10^{-4}, again, 5 instead of 10 Turbo iterations are sufficient. Furthermore, the modified system possesses a much higher convergence speed in both cases since it is almost converged after 10 iterations. In contrast to that, the conventional LTE system requires approximately 30 iterations. In (c) and (d), the convergence behavior is analyzed for both systems at a target code rate $r_{\mathrm{RM}} = 1/5$ (50 % dummy bits) and at channel SNRs

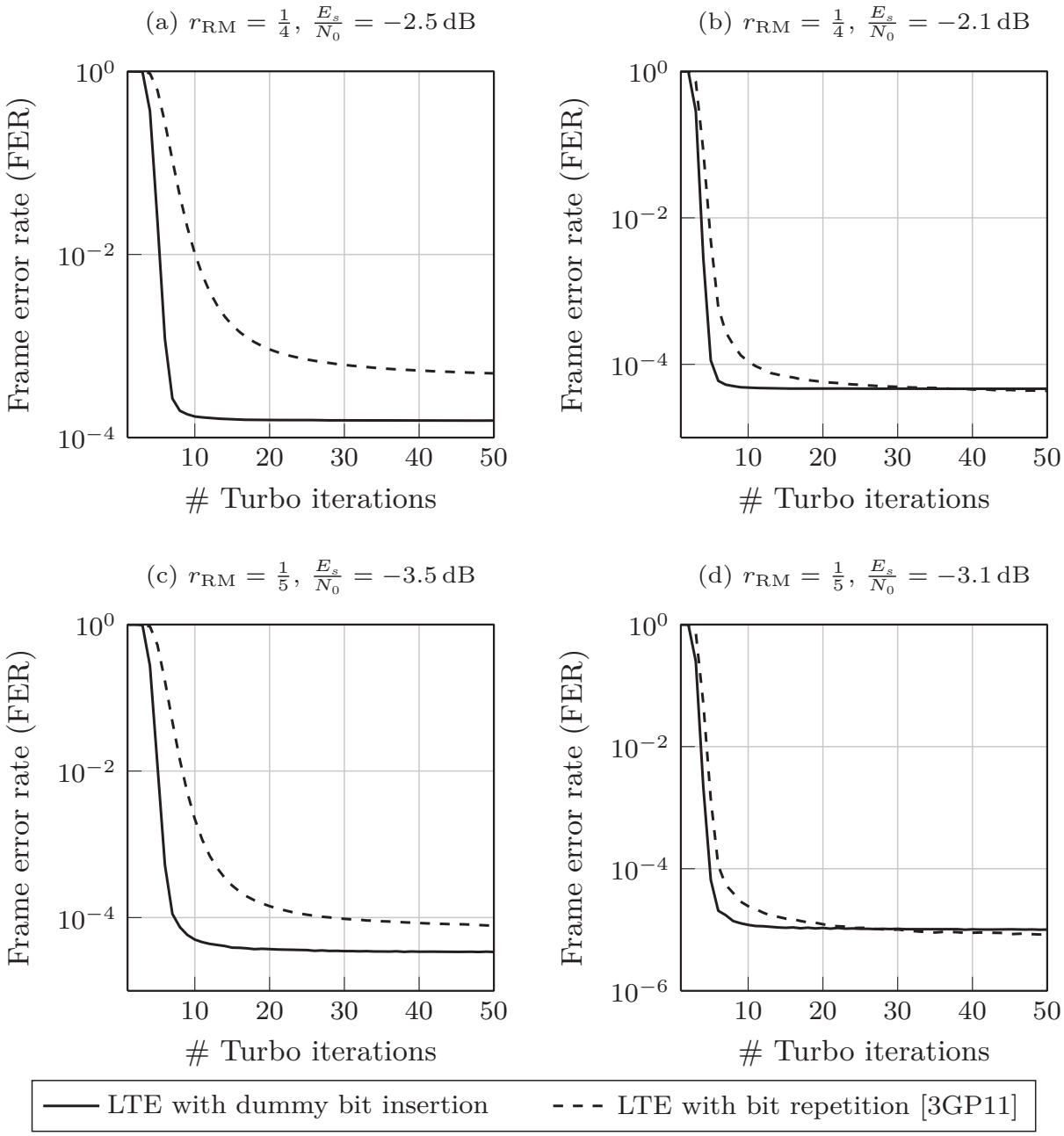


Figure 5.3: Convergence behavior of the modified LTE system using rate matching by dummy bit insertion compared to the conventional LTE system employing rate matching by bit repetition.

$E_s/N_0 = -3.5\,\mathrm{dB}$ (waterfall region) and $E_s/N_0 = -3.1\,\mathrm{dB}$ (error floor region), respectively. In both cases, less Turbo iterations are required for the modified LTE system to provide a desired target FER and, furthermore, a significantly increased convergence speed has been observed.

Finally, both systems are compared in terms of their throughput which is defined for a system without retransmissions and a specific modulation and coding scheme (MCS) according to

$$\mathcal{G}_{\mathrm{mcs}} = (1 - \mathrm{FER}) \cdot r_{\mathrm{RM}} \cdot I \tag{5.1}$$

with I denoting the number of bits per modulation symbol (e.g., $I = 2$ for QPSK) and FER the residual frame error rate. The maximum throughput $\mathcal{G}$ of the conventional (dashed lines) and the modified (solid lines) rate matching systems is depicted against the channel SNR E_s/N_0 in Figure 5.4. The communication channel is modeled by both, the Typical Urban (TU) channel with 20 MHz subchannel bandwidth specified by 3GPP (see Section A) and a simple AWGN channel. Since bit repetition and dummy bit insertion can only provide target code rates of $r_{\mathrm{RM}} < 1/3$, the system throughput is measured for the code rates

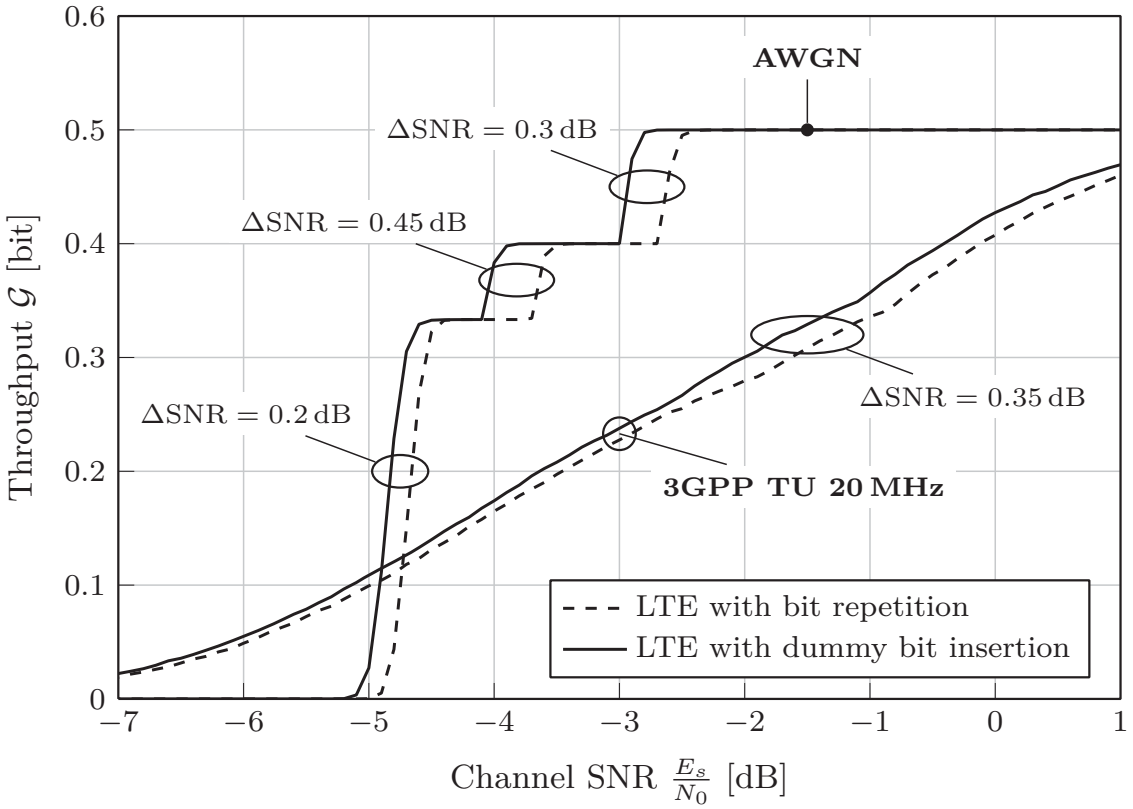


Figure 5.4: System throughput of the conventional LTE system employing bit repetition and the modified system using dummy bit insertion. Specified code rates after convolutional Turbo coding and rate matching: $r_{\mathrm{RM}} \in \{1/4, 1/5, 1/6\}$.

$r_{\mathrm{RM}} \in \{1/4, 1/5, 1/6\}$. Furthermore, both systems carry out 10 Turbo iterations. Gains up to $\Delta\mathrm{SNR} = 0.45\,\mathrm{dB}$ in the AWGN case and gains up to $\Delta\mathrm{SNR} = 0.35\,\mathrm{dB}$ for the 3GPP TU 20 MHz scenario are realized over a wide range of channel conditions.

It can be concluded from the above evaluation that dummy bit insertion used in conjunction with the LTE Turbo code provides in the extreme cases either notable performance gains in terms of FER and system throughput, respectively, or a significant convergence speedup saving decoding complexity and delay.

5.2 WIMAX

In IEEE 802.16e (WiMAX) [IEE04], rate matching is carried out by an ensemble of different LDPC codes, each optimized for one specific target code rate. Such a realization of adaptive coding suffers from a limited code rate flexibility and an increased implementation effort, since each LDPC code necessitates an individual encoder/decoder pair. Motivated by the common realization of rate-compatible Turbo codes, many techniques have been proposed in recent years to construct LDPC codes of higher or lower code rate from a fixed-rate mother LDPC code.

In [LN02], Li et al. have studied parity bit puncturing to construct LDPC codes of higher code rates. They have analyzed that this method only works well for high-rate LDPC codes and a small fraction of punctured bits. Therefore, they have proposed a special code extension technique to achieve lower code rates. Another approach of constructing codes of lower code rate from a high-rate mother LDPC code has been presented in [TJ05] using information shortening which is an alternative name for dummy bit insertion. They have proposed to select the dummy bit positions according to optimized degree distributions in order to achieve optimal threshold performance also for the shortened code. In [Oka08], this concept has further been applied to HARQ schemes. Incremental redundancy versions for additional HARQ transmissions are generated using sub-codes with optimized degree distributions, which have been constructed based on a high-rate mother LDPC code. The latter two methods attempt to improve the threshold performance of the rate-compatible LDPC code, since it is fully determined by the underlying degree distributions. However, only the number of dummy bits which have to be placed at variable nodes of a specific degree is determined by the derived target degree distribution. Note that they do not provide an optimized decision

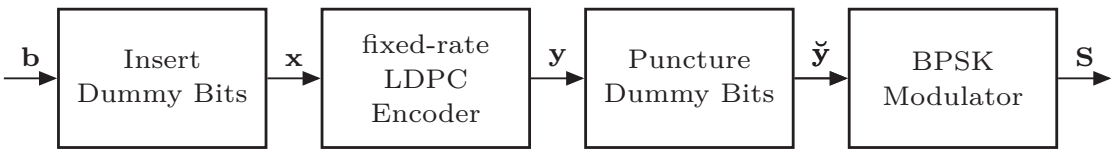


Figure 5.5: Rate matching by dummy bit insertion for WiMAX.

rule for the actual selection of dummy bit positions.

Therefore, a number of algorithms have been proposed in literature which attempt to insert dummy bits at exact positions. Many contributions focus on the sub-optimal *Belief Propagation* (BP) decoder and use dummy bit insertion to improve its decoding performance and/or convergence speed. Liu et al., e.g., presented in [LWZ09] an algorithm that selects dummy bit positions according to the variable node degrees and the so-called *extrinsic-sum* which is an artificial measure for the amount of *extrinsic* information received by a distinct variable node. Furthermore, in [BV13] dummy bits have been used to break cycles within the tanner graph which leads to a considerable increase in decoding performance.

The following section also concentrates on rate matching by dummy bit insertion and proposes an alternative rule to select heuristically optimized positions to insert known dummy bits into the information bit sequence. This approach has been pre-published in [BBV11] and follows the argumentation known from convolutional codes, where dummy bits are inserted equidistantly into the information bit vector before convolutional encoding. While convolutional codes introduce strong dependencies between neighboring bits, the relation between code bits of an LDPC code is not determined by the positions within a frame but solely by the connections of nodes in the Tanner graph. Thus, distributing variable nodes with perfect knowledge (dummy nodes) throughout the Tanner graph as uniformly as possible is not as trivial as for convolutional codes.

Optimized Dummy Bit Insertion

The following greedy heuristic is proposed to maximize the average minimum pairwise distance (i.e., the length of the shortest path) between a given number ℓ_d of dummy nodes (variable nodes associated with dummy bits) in a Tanner graph. First, a simple breadth first search is applied to find the minimum pairwise distances between all ℓ_x information nodes (variable nodes associated with information bits). The result is stored in a $\ell_x \times \ell_x$ distance matrix $\mathbf{D}$, where the matrix entry D_{ij} holds the minimum distance between information node at position i and information node at position j. Note that the distance matrix $\mathbf{D}$ is symmetric and has zeros on the main diagonal, since the Tanner graph is undirected and, thus, only $0.5 \cdot \ell_x \cdot (\ell_x - 1)$ elements (either the ones above or the ones below the main diagonal) have to be stored in memory.

After the distance matrix has been determined, ℓ_d information nodes are selected and added to the set of dummy nodes $\mathbb{V}$ in a node-by-node manner. The first two nodes that are added to $\mathbb{V}$ are chosen as $i_{\max}$ and $j_{\max}$ with

$$(i_{\max}, j_{\max}) = \arg\max_{\forall i,j} D_{ij}, \tag{5.2}$$

i.e., the two most distant information nodes. From the third node on, the node to be selected next is the information node i^* that maximizes the average distance to

all information nodes j that have already been selected:

$$i^* = \underset{\forall i \notin \mathbb{V}}{\arg\max} \frac{1}{|\mathbb{V}|} \sum_{\forall j \in \mathbb{V}} D_{ij}, \tag{5.3}$$

where $|\mathbb{V}|$ denotes the cardinality of $\mathbb{V}$. The algorithm stops when ℓ_d nodes have been added to $\mathbb{V}$ (i.e., $|\mathbb{V}| = \ell_d$). A summary of the algorithm is given in Algorithm 5.1.

For a rate matching scenario, the number of dummy bits ℓ_d is typically not fixed but possibly varies from one frame to the next. However, in this case it is still sufficient to run the algorithm once using $\ell_d = \ell_x$ and keep track of the order in which the information nodes are added to $\mathbb{V}$. This results in an ordered list of potential dummy bit positions which is of length ℓ_x. For each requested code rate smaller then the original code rate, the corresponding number of dummy bit positions $\ell'_d \leq \ell_x$ can then simply be obtained as the first ℓ'_d entries of this list.

Figure 5.6 shows an exemplary Tanner graph with five parity nodes and five information nodes three of which are used for dummy bits. In case (a), the dummy bit positions are simply chosen as the last three information bit positions resulting in an average pairwise distance of 2.67 edges between nodes associated with dummy bits. In case (b), on the other hand, the first, second, and fourth node are selected as dummy bit positions according to the proposed algorithm resulting in an increased average pairwise distance of 4 edges.

Simulation Results

In order to evaluate the performance of the proposed rate matching approach, the LDPC codes specified in the IEEE 802.16e WiMAX standard [IEE04] are considered. After LDPC encoding of $\ell_x = r_{\mathrm{RM}} \cdot \ell_y$ equiprobable information bits, BPSK modulation is applied and the resulting symbols are transmitted over an

Algorithm 5.1 Find K_d Optimized Dummy Bit Positions

1: Compute distance matrix $\mathbf{D}$ holding the minimum pairwise distances between all ℓ_x information nodes

2: Find two most distant nodes $(i_{\max}, j_{\max}) = \underset{\forall i,j}{\arg\max}\ D_{ij}$

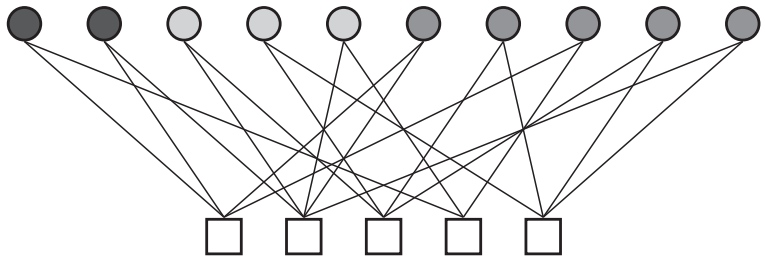
3: Initialize $\mathbb{V} = \{i_{\max}, j_{\max}\}$

4: **while** $|\mathbb{V}| < \ell_d$ **do**

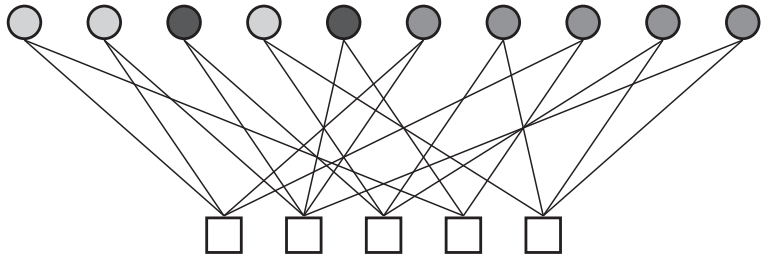
5: Find $i^* = \underset{\forall i \notin \mathbb{V}}{\arg\max}\ \frac{1}{|\mathbb{V}|} \sum_{\forall j \in \mathbb{V}} D_{ij}$

6: $\mathbb{V} = \mathbb{V} \cup \{i^*\}$

7: **end while**



(a) Simple code shortening: average pairwise distance of 2.67 edges between dummy nodes



(b) Optimized dummy bit insertion according to Algorithm 5.1: average pairwise distance of 4 edges between dummy nodes

Figure 5.6: Exemplary Tanner graph with five parity nodes and five information nodes, three of which are used for dummy bits. In case (a) simple code shortening is applied while case (b) uses Algorithm 5.1 to select dummy bit positions.

AWGN channel. The receiver uses a soft demapper followed by Belief Propagation decoding [Pea88] with a maximum of 200 iterations.

The following three setups are considered:

- **WiMAX code ensemble (dashed lines)**: Coded block length $\ell_y = 576$ and code rates $r_{\mathrm{RM}} \in \{1/2, 2/3, 3/4, 5/6\}$ as specified in the WiMAX standard [IEE04]. These codes are used as reference.

- **Simple-DBI (dotted lines)**: Dummy bits are simply appended at the end of the payload. The WiMAX LDPC code with code rate $r = 5/6$ is used as mother code and 40 %, 60 %, and 80 % of the available information bit positions are used for dummy bits to match the effective code rates after rate matching of $r_{\mathrm{RM}} = 3/4$, $r_{\mathrm{RM}} = 2/3$, and $r_{\mathrm{RM}} = 1/2$, respectively.

- **Optimized-DBI (solid lines)**: Same as for the Simple-DBI setup, but dummy bits are inserted at optimized positions according to Algorithm 5.1.

The block lengths ℓ_y of the codes for the Simple-DBI and the Optimized-DBI setups has been chosen to result in the same effective block length

$$\ell_y^{\mathrm{RM}} = \frac{1-r}{1-r_{\mathrm{RM}}} \cdot \ell_y = 576 \tag{5.4}$$

of the transmitted frame (payload and parity bits without dummy bits) as for the reference codes (i.e., $\ell_y = 864$ for $r_{\mathrm{RM}} = 3/4$, $\ell_y = 1152$ for $r_{\mathrm{RM}} = 2/3$ and $\ell_y = 1728$ for $r_{\mathrm{RM}} = 1/2$). Note that despite the increased block lengths of the codes using dummy bit insertion, decoding complexity is not increased since all additional nodes and edges of the Tanner graph are associated with dummy bits. These nodes and edges do not have to be considered during decoding, since they hold messages with values $+\infty$ which is the identity element of the box-plus operator [HOP96].

The upper plot in Figure 5.7 shows the frame error rate performance over the channel quality E_s/N_0 in dB. For an effective code rate after rate matching of $r_{\mathrm{RM}} = 3/4$, the waterfall behavior of all three setups is nearly identical, yet, the error floor performance is slightly degraded for the Simple-DBI setup and slightly enhanced for the Optimized-DBI setup. Except for a small loss of $\Delta\mathrm{SNR} < 0.1\,\mathrm{dB}$ in the waterfall region, a similar behavior is observed for an effective code rate of $r_{\mathrm{RM}} = 2/3$. The most evident difference between the Simple-DBI setup and the Optimized-DBI setup can be seen for the code rate $r_{\mathrm{RM}} = 1/2$. For the Simple-DBI case, the error floor already starts at a frame error rate of about 10^{-2}, in contrast to the Optimized-DBI case which comes very close to the original rate-1/2 WiMAX LDPC code with a loss of less than 0.2 dB near the waterfall and even less in the error floor region. It is evident that the WiMAX LDPC codes provide the best FER performance since they are optimized for the respective code rate. In modern wireless communication systems, however, a flexible and easy to implement code rate adaptation is indispensable in order to adapt the communication system to the instantaneous channel quality. According to Figure 5.7, rate matching by dummy bit insertion has shown tolerable performance losses in the evaluated scenarios, but provides a much higher flexibility in terms of code rate adaptation and is, thus, highly suited for wireless communication systems.

In the three lower plots of Figure 5.7, the convergence behavior is depicted for the three different setups. For each effective code rate r_{RM}, the FER is plotted over the number of decoding iterations for a fixed E_s/N_0 value. For $r_{\mathrm{RM}} = 1/2$ and $E_s/N_0 = 0.99\,\mathrm{dB}$ (left plot) it can be seen that convergence is much slower for the Simple-DBI setup than for the reference codes and the Optimized-DBI setup. The setup using optimized dummy bit insertion (Optimized-DBI), however, converges even faster than the original rate-1/2 WiMAX LDPC code but causes a slightly higher FER if the number of decoding iterations is high. For $r_{\mathrm{RM}} = 2/3$ and $E_s/N_0 = 2.64\,\mathrm{dB}$ (center plot), a very similar behavior is observed. For $r_{\mathrm{RM}} = 3/4$ and $E_s/N_0 = 3.75\,\mathrm{dB}$ (right plot), the original rate-3/4 WiMAX LDPC code exhibits the slowest convergence of all setups but still converges to a lower FER than

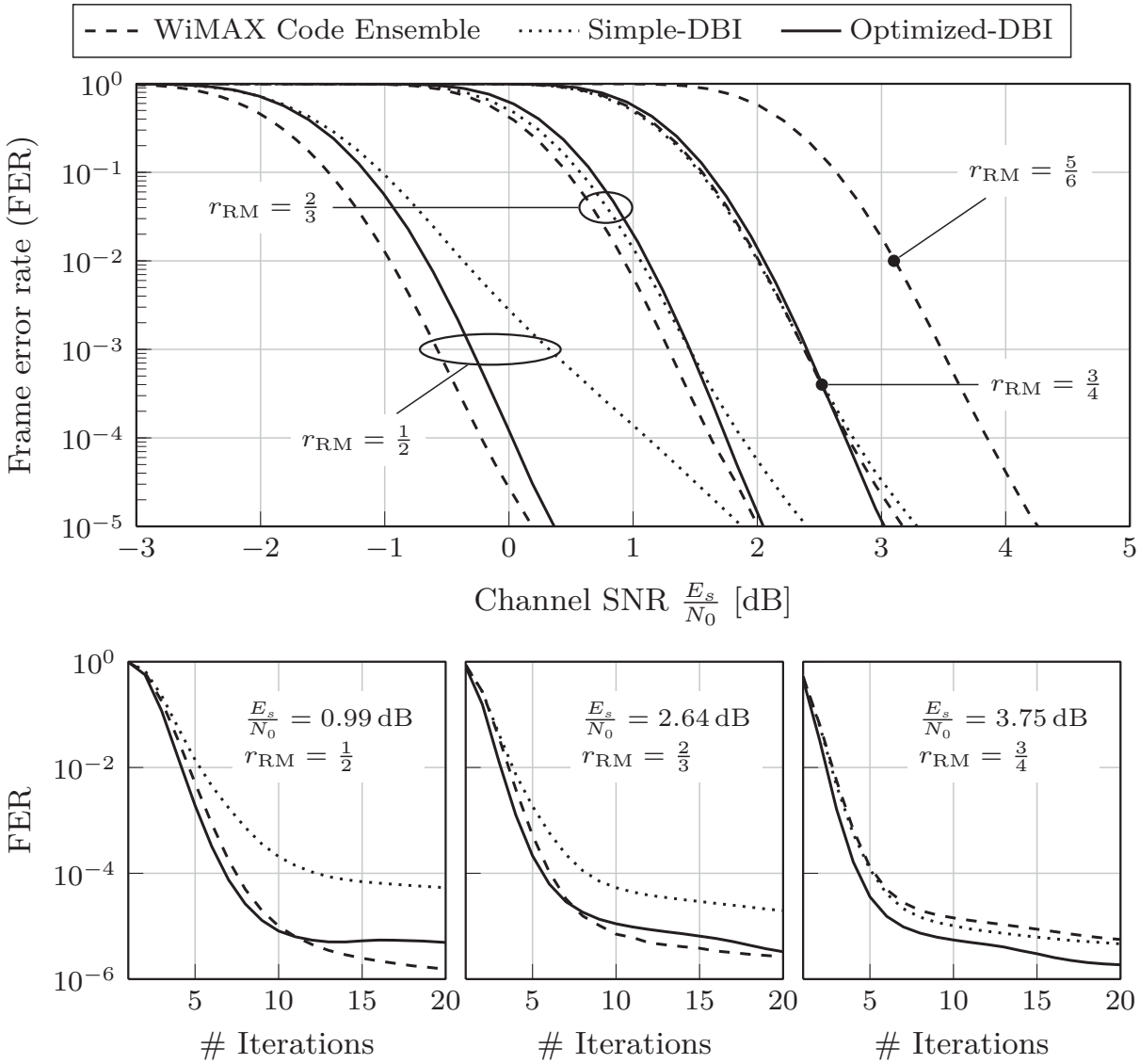


Figure 5.7: Simulation results for the WiMAX LDPC codes of effective block length $\ell_y^{\mathrm{RM}} = 576$. The upper plot shows the frame error rate (FER) over the channel quality E_s/N_0 after 200 decoding iterations. The lower plot shows the FER over the number of decoding iterations at fixed E_s/N_0 values.

for the Simple-DBI setup. The optimized dummy bit insertion (Optimized-DBI), however, has the fastest convergence and also achieves the best FER performance for small as well as for high numbers of iterations.

Chapter 6

Hybrid ARQ in Mobile Communication Systems

Hybrid Automatic Repeat reQuest (HARQ) schemes [LC82] are conventionally employed in modern wireless and mobile communication systems [3GP11] to reduce the effect of channel degradations on the system performance. *Forward Error Correction* (FEC) coding as, e.g., Turbo or LDPC coding is used in these schemes additionally to standard ARQ. In standard ARQ, only *Cyclic Redundancy Check* (CRC) bits are attached to the frames before transmission and error detection is performed at the receiver based on this CRC. If the CRC detects an erroneous frame, retransmissions of the same frames are requested by the receiver. The corresponding signaling is carried out over a feedback channel by means of *ACKnowledge* (ACK) and *Negative ACKnowledge* (NACK) signals. Accordingly, ARQ schemes provide implicitly adaptive coding by simple frame repetition. In HARQ schemes, FEC coding is additionally employed in order to reduce the overall number of (re)transmissions and, thus, the average signal delay. It can be distinguished between three basic types of HARQ schemes:

- **HARQ type-I [LC82]**: Each encoded data frame is retransmitted until the frame passes the CRC or the maximum number of retransmissions is reached (truncated HARQ). Faulty decoded frames are directly discarded, i.e., their information is not additionally exploited at the receiver.

- **HARQ type-II [Man74]**: In contrast to HARQ type-I, each transmission contains *Incremental Redundancy* (IR) about the data frame. The so-called *Redundancy Version*s (RVs) are then combined at the receiver before channel decoding. This method is also known as HARQ with incremental redundancy or shortly IR-HARQ.

- **HARQ type-III [Cha85]**: This type is a combination of HARQ type-I and HARQ type-II. Each transmission contains the same redundancy about the data frame. But in contrast to HARQ type-I, all transmissions are weighted by the current channel SNR and combined at the receiver before channel decoding. In literature, this method is also referred to as HARQ type-II with chase combining or HARQ type-II with code combining.

According to Section 3.4, HARQ type-II is employed in LTE. Incremental redundancy is generated by bit puncturing and bit repetition based on a simple ring buffer implementation after rate-1/3 Turbo coding. However, *Rate-Compatible Insertion Convolutional* (RCIC) Turbo codes, which are based on dummy bit insertion, have been proven to provide significant performance gains in terms of residual *Frame Error Rate* (FER) compared to the conventional LTE rate matching for code rates $r_{\mathrm{RM}} < 1/3$ (see Section 5.1). Consequently, dummy bit insertion seems to be a promising alternative to simple bit repetition. However, in systems employing HARQ type-II, i.e., where *incremental redundancy* transmissions can be carried out, the full exploitation of the potential of RCIC codes is more challenging. Dummy bits will only be inserted before Turbo encoding if a target code rate which is lower than the code rate of the employed Turbo code is desired for the initial transmission. Higher code rates are achieved without dummy bit insertion by bit puncturing. In case of decoding failures, additional transmissions are requested by the receiver lowering the effective code rate after rate matching and HARQ. However, even if the effective code rate after HARQ drops below the code rate of the employed Turbo code, the advantages of the dummy bit insertion cannot be exploited. This is due to the fact that no code rate adaptation can be performed during HARQ by means of dummy bit insertion since the fraction of inserted dummy bits is fixed after Turbo encoding. Therefore, e.g., bit repetition has to be performed instead which results in the conventional LTE rate matching scheme.

In this chapter, a novel HARQ scheme for LTE based on RCIC codes will be proposed which copes with the above mentioned performance limitation (Section 6.1). This contribution has been pre-published in [BEV12a] and comprises

- a scheme which exploits the potential of RCIC codes as soon as the effective code rate after rate matching and HARQ drops below the code rate of the fixed rate Turbo code. This code rate does not have to be known before Turbo encoding since the fraction of effective dummy bits can be adapted accordingly during the HARQ process,
- a low complexity implementation based on a modified LTE ring buffer,
- a new dummy bit interleaver which enables the optimal exploitation of the dummy bits at the decoder.

In the second part of this chapter, the influence of an unreliable feedback channel on the performance of HARQ type-II schemes is addressed. So far, the influence of HARQ schemes on the system performance has mostly been studied for reliable feedback channels without feedback errors, e.g., in [Kal90, CF05, CCY05, Che06, LMS07, LLMC12] which cannot be assumed in general. In [ZS05], the impact of signaling errors on the throughput of *High Speed Uplink Packet Access* (HSUPA) is verified by simulations. In [SM09], the coverage of LTE is improved

by reducing the impact of ACK/NACK signaling errors using a technique which is called *Transmission Time Interval* (TTI) bundling. However, there are only a limited number of contributions which examine analytically the impact of unreliable ACK/NACK feedback. In most of the publications, the analysis is done for special HARQ protocols or channel models. In [ZR95a, ZR95b] for example, the performance of the ARQ Go-Back-N protocol and in [AN07, Bad09] the effectiveness of the selective repeat protocol in Markov channels is analyzed under the assumption of unreliable feedback. Furthermore, the impact of noisy feedback on various types of HARQ schemes is discussed in detail for block fading channels in [ML00, EBBS05, Bou10, Wu10, MCM11]. It is further assumed in most contributions that NACK→ACK errors can be detected by a CRC and, thus, be avoided by the considered communication system. However, this might not be the case in a real mobile communication system such as, e.g., *High Speed Packet Access* (HSPA) [3GP06], in which error handling strategies are formulated for ACK→NACK as well as NACK→ACK errors.

Therefore, a more general analysis of HARQ systems with unreliable feedback will be provided in Section 6.2 according to the derivation presented in [BEV12b]:

- We will focus on a general system model which is not linked to a special communication channel and/or HARQ protocol.
- No restrictions are made for the error handling of ACK/NACK signaling errors, i.e., NACK→ACK errors are included into the analysis.
- Expressions for the required number of transmissions per frame and the resulting overall system throughput are derived analytically in dependency of the residual *Frame Error Rates* (FERs) and the current quality of the feedback channel.
- Links to specific cases are given which can be covered by our general analysis.

Based on the residual frame error rates of the communications system with reliable feedback, the influence of signaling errors can now be evaluated analytically supporting the design of feedback transmission schemes (e.g., required code rate, transmission power). Time-consuming simulations of the respective system with unreliable feedback can, thus, be avoided. The correctness and the benefit of the presented analysis is verified by extensive LTE physical layer simulations.

6.1 Novel HARQ for LTE Based on RCIC Codes

As shown in Figure 5.2, the application of RCIC Turbo codes instead of *Rate-Compatible Repetition Convolutional* (RCRC) Turbo codes, as in LTE, provides considerable gains in the waterfall region in terms of residual FER for code rates of $1/5 \leq r_{\mathrm{RM}} < 1/3$ after convolutional Turbo coding and rate matching. Using such

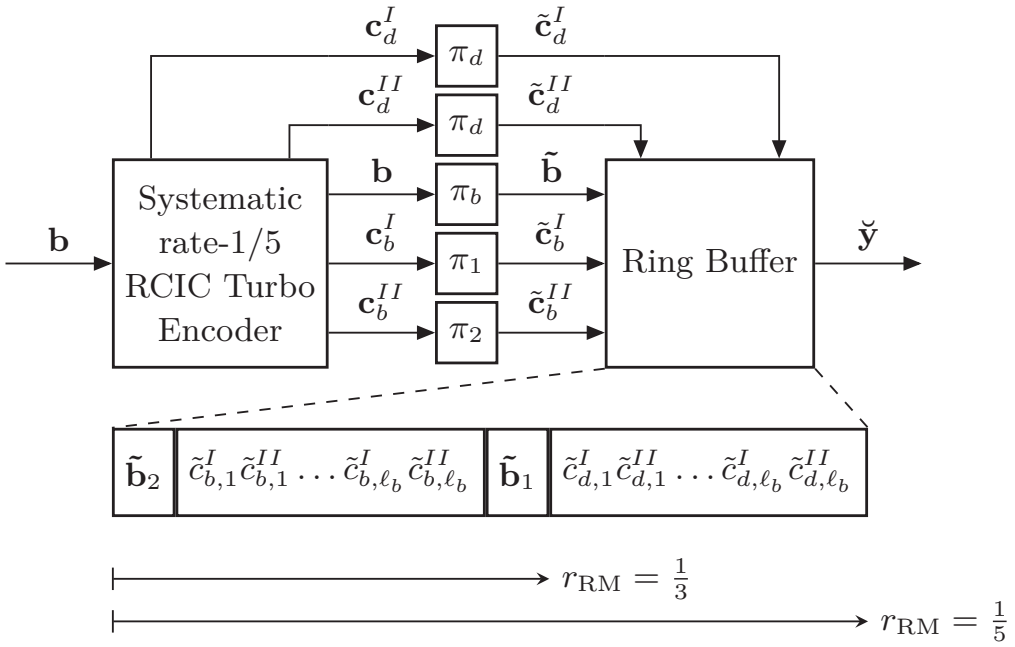


Figure 6.1: Low rate RCIC Turbo code with ring buffer implementation used in the novel HARQ scheme.

a code in a conventional HARQ scheme is of minor benefit since these gains can only be realized if the code rate selected for the initial transmission lies in the above mentioned range. If code rates $r_{\mathrm{RM}} \geq 1/3$ (no dummy bits) are initially chosen and additional transmissions are requested due to decoding errors, no benefit will be expected although the effective code rate after rate matching and HARQ may be within the desired range. In this case, only bit repetition is employed and the modified system reduces to the conventional one since the fraction of dummy bits cannot be adapted during HARQ.

Therefore, a novel HARQ system for LTE is proposed which benefits from the gains provided by dummy bit insertion as soon as the effective code rate after rate matching and HARQ drops below the threshold $r_{\mathrm{RM}}/\rho = 1/3$ (ρ is the number of transmissions). Furthermore, an effective ring buffer implementation, which is similar to the LTE realization, has been developed and will be described in what follows.

The structure of the novel rate matching unit is shown in Figure 6.1. The information bit vector $\mathbf{b}$ of length ℓ_b is encoded by an RCIC Turbo encoder with fixed code rate $r_{\mathrm{IC}} = 1/5$ which is the lowest useful code rate according to Figure 5.2. The LTE Turbo code is employed as mother code. Taking Algorithm 3.1 and Equation (3.5) into account, this results in an equidistant insertion of $\ell_d = \ell_b$ dummy bits. Five streams, each of length ℓ_b, are generated by the encoder: The systematic information bits $\mathbf{b}$, the parity bits $\mathbf{c}_b^I$ and $\mathbf{c}_b^{II}$ corresponding to the information bit positions and the parity bits $\mathbf{c}_d^I$ and $\mathbf{c}_d^{II}$ corresponding to the dummy bit positions. All streams are individually interleaved and written to the ring buffer according

to the structure depicted in Figure 6.1. The interleaved information bit vector $\tilde{\mathbf{b}}$ is separated into two streams $\tilde{\mathbf{b}}_1 = (\tilde{b}_1, \ldots, \tilde{b}_{\theta_1 - 1})$ and $\tilde{\mathbf{b}}_2 = (\tilde{b}_{\theta_1}, \ldots, \tilde{b}_{\ell_b})$ with $\tilde{\mathbf{b}} = [\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2]$ where θ_1 is determined by (3.6). The modified initial reading position θ'_ρ of a distinct RV ρ $(1 \leq \rho \leq K)$ is now given by

$$\theta'_\rho = \mathrm{mod}\left(((\rho - 1) \cdot \ell_{\breve{y}}), 5\,\ell_b\right) + 1, \tag{6.1}$$

where $\ell_{\breve{y}}$ is the length of each RV, $5\,\ell_b$ the size of the ring buffer, and $\mathrm{mod}\,(\cdot,\cdot)$ the modulo operation. Consequently, it can be concluded that

(a) the first RV starts at the beginning of the ring buffer,

(b) the next RV ρ starts at the end of the previous RV $\rho - 1$.

The special structure in conjunction with property (a) guarantees that the first RV of the modified systems equals the first RV of the conventional LTE system for any code rates $r_{\mathrm{RM}} \geq 1/3$ after rate matching. For code rates $1/5 \leq r_{\mathrm{RM}} \leq 1/3$, the RCIC decoder can benefit from the dummy bit insertion since a fraction of parity bits $\mathbf{c}_d^I$ and $\mathbf{c}_d^{II}$ are additionally transmitted (cf. Figure 6.1). In order to provide effective code rates $r_{\mathrm{RM}}/\rho < 1/5$ after ρ transmissions, the ring buffer is continuously read out resulting in repetition of systematic and parity bits.

A key element in the novel HARQ scheme is the interleaver π_d which rearranges the position of the parity bits $\mathbf{c}_d^I$ and $\mathbf{c}_d^{II}$ within the ring buffer according to a bijective interleaver mapping $l \leftrightarrow \Gamma(l)$: $\tilde{c}^I_{d,\Gamma(l)} = c^I_{d,l}$ and $\tilde{c}^{II}_{d,\Gamma(l)} = c^{II}_{d,l}$. As described in Section 3.3, best performance can be expected if all bits which are read out for transmission are equidistantly distributed within the frame. Accordingly, a low complexity, recursive algorithm has been developed which can construct the required interleaver mapping for any interleaver size on the fly saving a lot of memory. The core algorithm only operates on block sizes 2^Λ, $\Lambda \in \mathbb{N}$. Therefore, if necessary, filler bits are appended to the vectors $\mathbf{c}_d^I$ and $\mathbf{c}_d^{II}$ before interleaving to match a block size of a power of two. These bits are removed again after interleaving. The core algorithm is given in Algorithm 6.1 and determines recursively an index vector $\mathbf{w}_\lambda$ $(0 \leq \lambda \leq \Lambda)$ for a block size 2^λ in each step λ. The vector $\mathbf{w}_\Lambda$

Algorithm 6.1 Dummy bit index vector $\mathbf{w}_\Lambda$ for block sizes 2^Λ

Initialize: $\mathbf{w}_0 = 1$
  for all λ such that $1 \leq \lambda \leq \Lambda$ **do**
    $\mathbf{w}_\lambda = [(2\mathbf{w}_{\lambda-1} - 1), 2\mathbf{w}_{\lambda-1}]$
  end for

contains the interleaver mapping for block sizes 2^Λ. The removal of the filler bits results in the vector $\mathbf{w}'_\Lambda = (w'_{\Lambda,1}, \ldots, w'_{\Lambda,l}, \ldots, w'_{\Lambda,\ell_d})$ which provides the desired interleaver mapping $\Gamma(l) = w'_{\Lambda,l}$, $1 \leq l \leq \ell_d$. For the sake of clarity, the following example is considered: An interleaver mapping for a block size of $\ell_d = 2^4 = 16$

bits ($\mathbf{w}'_\Lambda = \mathbf{w}_\Lambda$) shall be constructed by means of Algorithm 6.1. After the initialization step $\mathbf{w}_0 = 1$, the recursion generates stepwise the desired index vector $\mathbf{w}_4$:

$$\mathbf{w}_1 = (1, 2) \tag{6.2}$$

$$\mathbf{w}_2 = (1, 3, 2, 4) \tag{6.3}$$

$$\mathbf{w}_3 = (1, 5, 3, 7, 2, 6, 4, 8) \tag{6.4}$$

$$\mathbf{w}_4 = (1, 9, 5, 13, 3, 11, 7, 15, 2, 10, 6, 14, 4, 12, 8, 16). \tag{6.5}$$

Then rate matching is exemplarily performed with

(a) a code rate $r_{\mathrm{RM}} = 2/7$ resulting in the transmission of 1/4 of the parity bits $\mathbf{c}_d^I$ and $\mathbf{c}_d^{II}$. The effective dummy bit index vector is given by $\breve{\mathbf{w}}'_4 = (1, 9, 5, 13)$ in this case. Consequently, the effective dummy bits correspond to the index vector $\mathbf{w}'_4 = (1, 5, 9, 13)$ after ordering in ascending order and are, thus, equidistantly distributed.

(b) a code rate $r_{\mathrm{RM}} = 1/4$ resulting in the transmission of 1/2 of the parity bits $\mathbf{c}_d^I$ and $\mathbf{c}_d^{II}$. In this case, the effective dummy bit index vector is given by $\breve{\mathbf{w}}'_4 = (1, 9, 5, 13, 3, 11, 7, 15)$. Consequently, the effective dummy bits correspond to the index vector $\mathbf{w}'_4 = (1, 3, 5, 7, 9, 11, 13, 15)$ after ordering in ascending order and are equidistantly distributed again.

Consequently, the channel-related information $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{c}_d^I)$ and $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{c}_d^{II})$ provided to the corresponding RCIC decoder is equidistantly distributed. In these cases, the system performance will be equivalent to a system only consisting of a single RCIC code with fixed code rate $r_{\mathrm{IC}} = 2/7$ and $r_{\mathrm{IC}} = 1/4$, respectively, as it will be proven in the next section. This demonstrates that the specific construction rule of the interleaver allows to control the effective code rate after encoding without any performance degradation. This property is essential for the effectiveness of the proposed HARQ scheme. Furthermore, for any other fractions at least a homogeneous distribution of the dummy bits is provided. For example, a code rate $r_{\mathrm{RM}} = 2/9$ results in the transmission of 3/4 of the parity bits $\mathbf{c}_d^I$ and $\mathbf{c}_d^{II}$. The parity bits which are not selected for transmission correspond to the indexes $(4, 12, 8, 16)$ in this case and are, thus, equidistantly distributed. This behavior guarantees that at least a homogeneous distribution of dummy bits is provided for effective dummy bit fractions after rate matching of $\ell_d^{\mathrm{RM}}/\ell_b \neq 2^{-\lambda}$, $\lambda \in \mathbb{N}$.

Effect of Dummy Bit Puncturing

The flexibility in code rate adaptation is achieved by means of a high rate RCIC Turbo code and succeeding puncturing of a distinct number of information bits and parity bits. In the presented scheme, it has to be distinguished between the puncturing of parity bits which correspond to information bit positions and

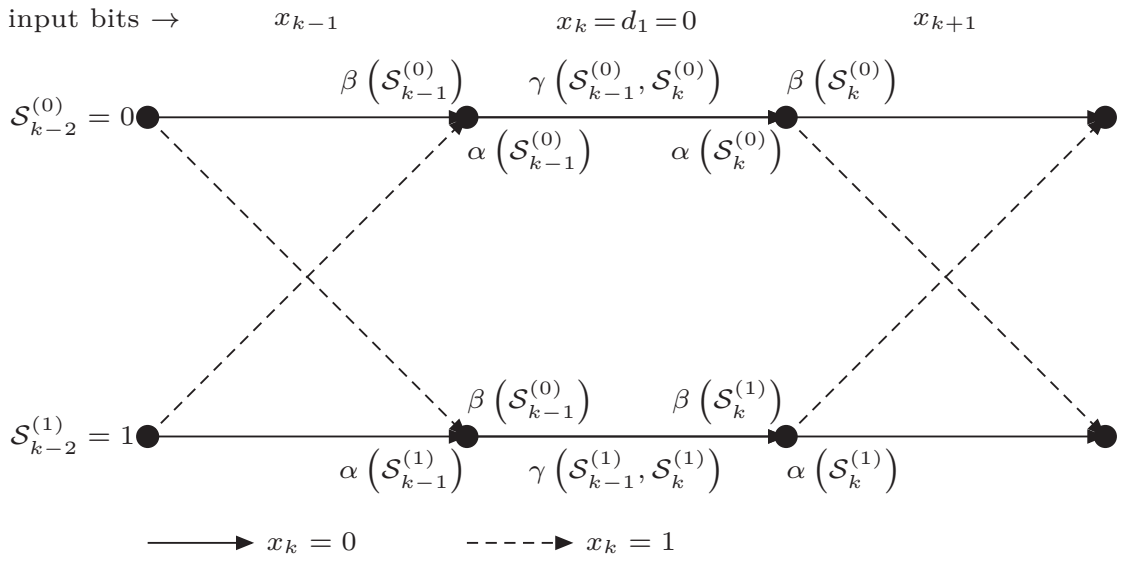


Figure 6.2: Trellis diagram for a rate-1 recursive convolutional code with octal generator polynomial $\mathbf{G}_{\mathrm{CC}} = (2/3)_8$. One dummy bit $d_1 = 0$ is inserted at time instance k.

parity bits which correspond to dummy bit positions. While the effect of the first puncturing is well studied in recent years, e.g., in [Hag88], the latter case, shortly referred to as *dummy bit puncturing*, is not analyzed yet. However, such an analysis is extremely beneficial in order to evaluate the effectiveness of the novel LTE HARQ scheme.

Each convolutional SISO component decoder receives channel-related LLR vectors $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{b})$, $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{c}_b)$ and $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{c}_d)$ for the systematic bits and all parity bits. Note that the superscripts are omitted in this case for the sake of clarity. Furthermore, perfect *a priori* information about the dummy bits is provided to the decoder. This results in a pruned trellis diagram as described in Section 3.3 and exemplarily depicted in Figure 6.2 for a rate-1 recursive convolutional code with octal generator polynomial $\mathbf{G}_{\mathrm{CC}} = (2/3)_8$. One dummy bit $d_1 = 0$ is inserted into the information bit sequence $\mathbf{b}$ at time instance k resulting in the input frame $\mathbf{x}$ of the considered convolutional code. The pruned trellis at each dummy bit position in conjunction with the additional channel-related information $\mathbf{L}(\mathbf{c}_d)$ can then be exploited during the *Bahl-Cocke-Jelinek-Raviv* (BCJR) [BCJR74] decoding of the information bits $\mathbf{b}$. Considering the trellis diagram of a convolutional code with constraint length $\mathfrak{L} + 1$ (cf. Figure 6.2, $\mathfrak{L} = 1$) at the dummy bit position k, half of the state transitions disappear. Furthermore, each state $\mathcal{S}_{k-1}^{(i)}$ $(0 \leq i < 2^{\mathfrak{L}})$ is connected with exactly one succeeding state $\mathcal{S}_k^{(j)}$ $(0 \leq j < 2^{\mathfrak{L}})$. The forward and backward recursion (cf. Appendix C) $\alpha\left(\mathcal{S}_k^{(j)}\right)$ and $\beta\left(\mathcal{S}_k^{(i)}\right)$ at time instance k can be expressed in the logarithmic domain for each valid transition $\mathcal{S}_{k-1}^{(i)} \rightarrow \mathcal{S}_k^{(j)}$

according to

$$\alpha\left(\mathcal{S}_k^{(j)}\right) = \gamma\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right) + \alpha\left(\mathcal{S}_{k-1}^{(i)}\right), \tag{6.6}$$

$$\beta\left(\mathcal{S}_{k-1}^{(i)}\right) = \gamma\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right) + \beta\left(\mathcal{S}_k^{(j)}\right). \tag{6.7}$$

According to Appendix C, the innovations $\gamma\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right)$ at a dummy bit position k for a systematic convolutional code are generally given by

$$\gamma\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right) = \frac{1}{2}\sum_{n\in\mathbb{P}} y_n L_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n) = \frac{1}{2}\sum_{n'\in\mathbb{P}'} c_{d,n'} L_{\mathrm{DM}}^{[\mathrm{chan}]}(c_{d,n'}) \tag{6.8}$$

exploiting the channel-related reliabilities $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{c}_d)$. $\mathbb{P}$ and $\mathbb{P}'$ denote the subsets of indexes which correspond to the parity bits that participate at time instance k with regard to $\mathbf{y}$ and $\mathbf{c}_d$, respectively. Note that the perfect knowledge of the dummy bit $d_1 = 0$ has already been taken into account by the pruned state transitions. It can be concluded from (6.6), (6.7), and (6.8) that RCIC codes basically gain from the additional channel-related information provided by the parity bits $\mathbf{c}_d$ generated at each dummy bit position. In other words, if $\mathbf{c}_d$ would be completely punctured, (6.6), (6.7), and (6.8) reduces to

$$\gamma\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right) = 0\,, \quad \alpha\left(\mathcal{S}_k^{(j)}\right) = \alpha\left(\mathcal{S}_{k-1}^{(i)}\right)\,, \quad \beta\left(\mathcal{S}_{k-1}^{(i)}\right) = \beta\left(\mathcal{S}_k^{(j)}\right) \tag{6.9}$$

which implies that the effect of the dummy bits disappears without any loss in performance compared to the convolutional mother code. This behavior is extremely beneficial for the novel HARQ scheme. The main property can be summarized as follows:

> The effective code rate r_{RM} does not have to be known before RCIC encoding. It can be controlled after encoding without any loss in performance by just puncturing the respective number of parity bits $\mathbf{c}_d$. This ensures an extremely high flexibility in terms of code rate adaptation, since a sufficient number of dummy bits can be pre-inserted.

Evaluation

The conventional LTE HARQ system as well as the novel RCIC based LTE HARQ system have been simulated in an AWGN environment and a typical urban (TU) environment using the 3GPP TU channel model for a system bandwidth of 20 MHz. Simulations have been conducted with 10 Turbo iterations, a maximum of $K = 4$ transmissions, and with *Modulation and Coding Schemes* (MCSs) as indicated in Table 6.1, where I is the number of code bits per modulation symbol.

Table 6.1: Modulation and coding schemes (MCS).

Modulation	Code rates
QPSK ($I = 2$)	$r_{\mathrm{RM}} \in \{\frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}\}$
16QAM ($I = 4$)	$r_{\mathrm{RM}} \in \{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}\}$
64QAM ($I = 6$)	$r_{\mathrm{RM}} \in \{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}\}$

Both systems are compared in terms of their throughput which is defined for a specific MCS as

$$\mathcal{G}_{\mathrm{mcs}} = (1 - \mathrm{FER}) \cdot \frac{r_{\mathrm{RM}} \cdot I}{\overline{K}} \tag{6.10}$$

with $\overline{K}$ denoting the average number of required transmissions and FER the frame error rate after the last transmission. According to Table 6.1, 15 different throughput curves each corresponding to one specific MCS are obtained.

Figure 6.3 shows the envelope $\mathcal{G}$ of these throughput curves for the conventional LTE HARQ system based on bit puncturing and bit repetition (dashed lines) as well as for the novel LTE HARQ scheme (solid lines) based on bit puncturing and dummy bit insertion. Since a wide range of channel conditions is covered by both systems, the complete SNR range is divided into a lower SNR range (upper figure) and a higher SNR range (lower figure). The Shannon bound is also depicted in both figures as reference (dash dotted lines). In the AWGN environment, dummy bit insertion is most effective in the lower SNR range. For these channel conditions, the novel approach performs better than or equal to the conventional LTE HARQ scheme with gains up to $\Delta\mathrm{SNR} = 0.7\,\mathrm{dB}$ reducing the gap to the Shannon bound. In the higher SNR range, the dummy bit insertion has less effect on the maximum throughput $\mathcal{G}$ since MCS modes comprising higher code rates (1/2, 2/3, 3/4, 4/5) are mainly employed. Consequently, improvements of the system throughput can only be expected if the maximum throughput is provided by such an MCS after carrying out additional transmissions that leads to an effective code rate lower than 1/3. Nevertheless, gains up to $\Delta\mathrm{SNR} = 0.95\,\mathrm{dB}$ have been achieved for the AWGN channel. Considering the typical urban scenario, improvements in system throughput can be realized within a wide rang of channel conditions ($-12\,\mathrm{dB} \leq E_s/N_0 \leq 13\,\mathrm{dB}$). Gains up to $\Delta\mathrm{SNR} = 1.4\,\mathrm{dB}$ have been observed.

In Figure 6.4 the throughput for three distinctive modulation and coding schemes is depicted exemplarily for the AWGN simulation. Again, the gains of the novel LTE HARQ scheme are obvious: for more than one transmission, the novel system provides a much better performance, e.g., $\Delta\mathrm{SNR} = 1.5\,\mathrm{dB}$ for 64QAM with code rate $r_{\mathrm{RM}} = 1/2$ for each RV. The novel HARQ scheme benefits from the dummy bit insertion method as soon as the effective code rate r_{RM}/ρ after rate matching and HARQ drops below the code rate of the mother code ($r_{\mathrm{CC}} = 1/3 \Rightarrow$ gain for $\rho \geq 2$).

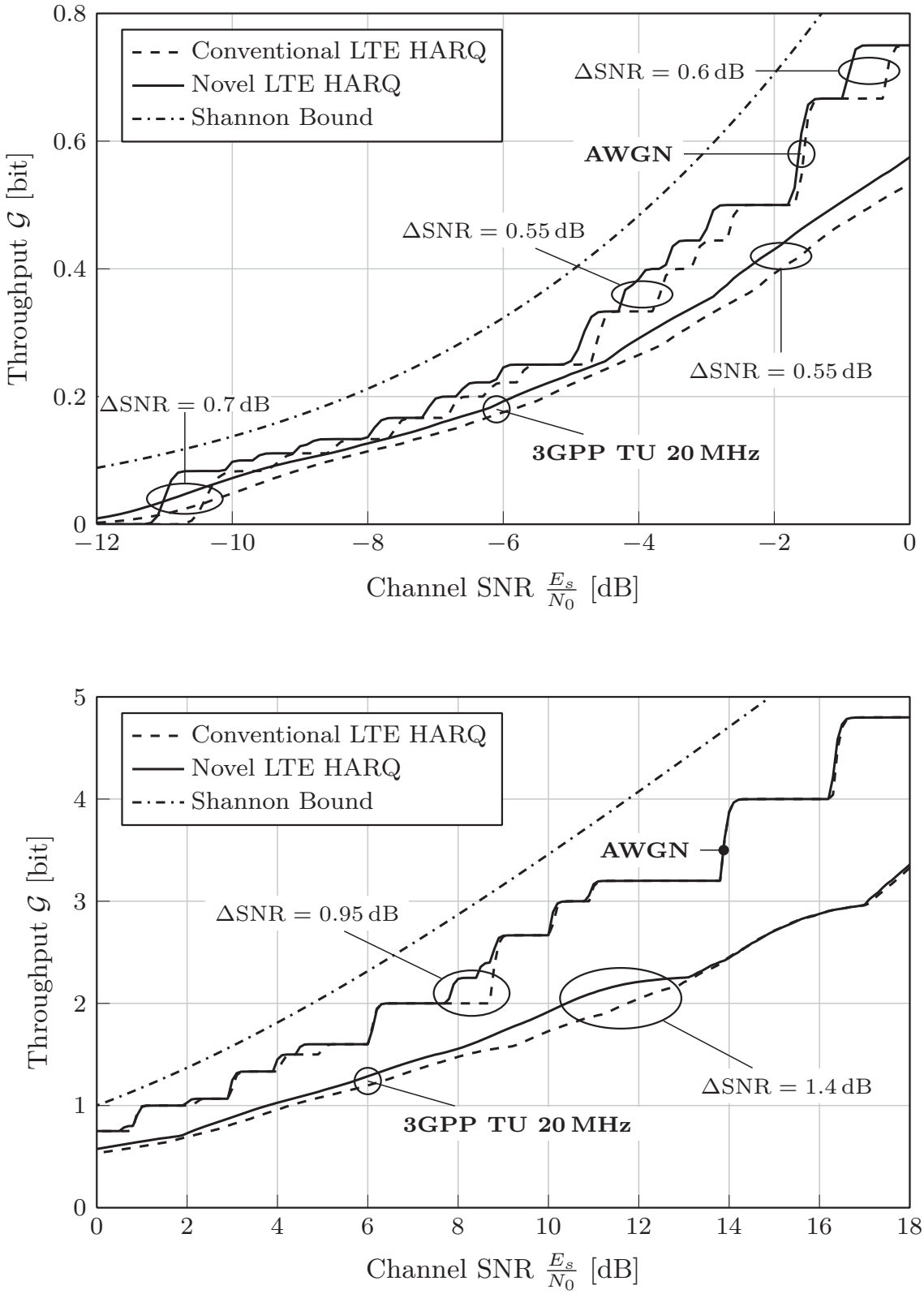


Figure 6.3: Maximum system throughput (envelope) of the novel LTE HARQ system compared to the conventional LTE HARQ system. Comparison is performed for a transmission over an AWGN channel and the typical urban (TU) channel with system bandwidth of 20 MHz specified by 3GPP.

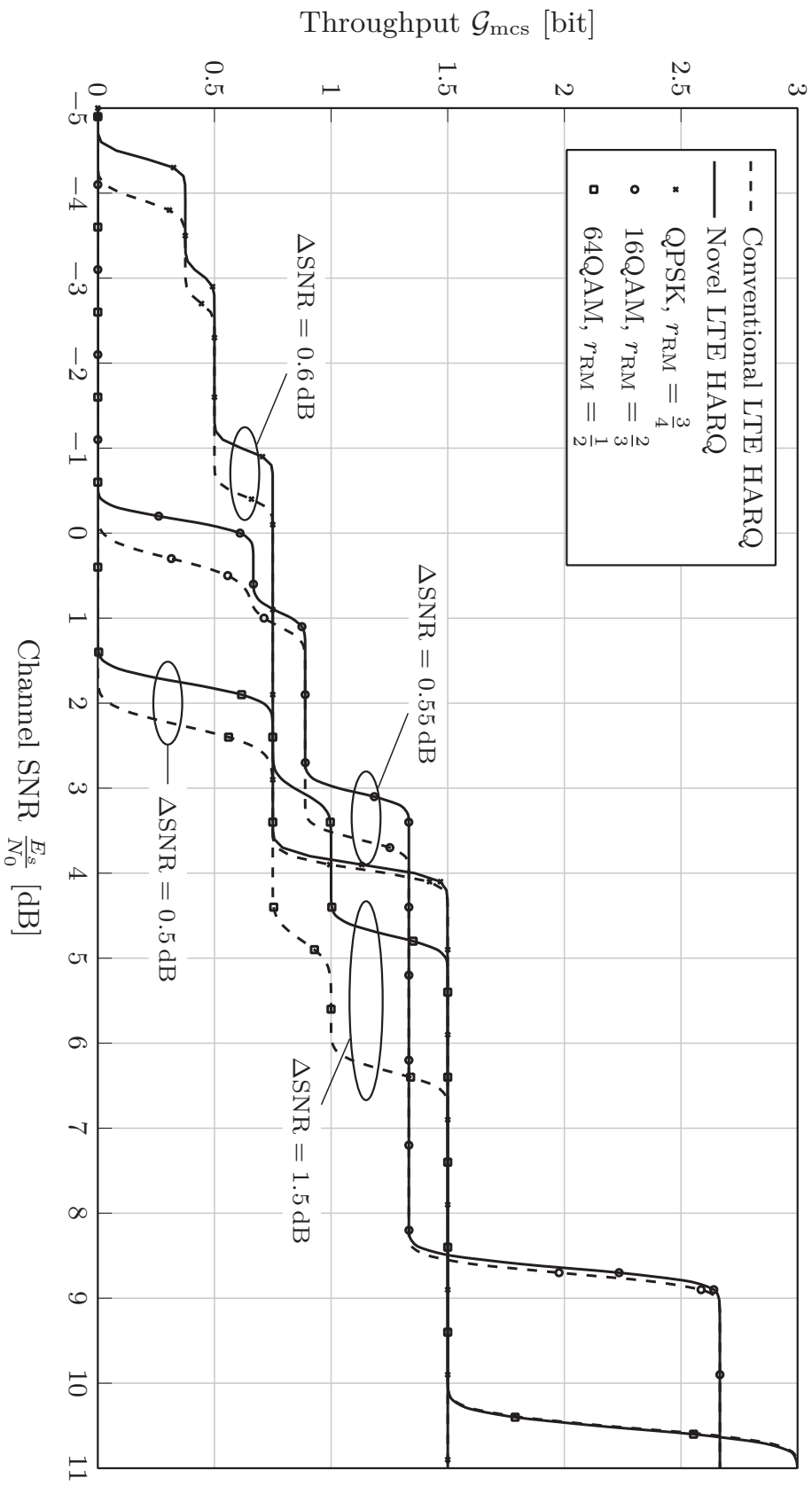


Figure 6.4: System throughput of the novel LTE HARQ system compared to the conventional system for three different modes.

This is the major advantage compared to RCIC systems known from literature, in which the code rate after encoding and rate matching has to be known at the RCIC encoder, i.e., before HARQ has been carried out. These systems cannot adapt their code rate by dummy bit insertion afterwards if additional RV transmissions are requested by the receiver. In particular for systems without perfect channel state information, the novel LTE HARQ system outperforms the conventional one. In this case, additional RV transmissions have to be carried out more frequently resulting in higher throughput gains. Furthermore, the LTE-like ring buffer structure simplifies the integration of the proposed HARQ scheme in future mobile communication systems since only minor transceiver modifications are required.

6.2 HARQ With Unreliable Feedback

In most evaluations regarding the system throughput of wireless or mobile communication systems employing HARQ, a perfect feedback channel for the transmission of ACK/NACK frames is assumed. In particular in wireless systems, however, various impairments on the physical link reduce the reliability of the feedback transmission. In [BEV12b] we have shown that the influence of an unreliable feedback channel on the system throughput of an arbitrary digital communication system can be analytically expressed. This derivation is revisited in what follows.

A general model for an unreliable feedback channel which covers ACK→NACK as well as NACK→ACK signaling errors is depicted in Figure 6.5. This model is defined by two conditional error probabilities $P_\mathrm{A} = \mathrm{Prob}\{\mathcal{W}_i = \mathrm{NACK}|\mathcal{V}_i = \mathrm{ACK}\}$ and $P_\mathrm{N} = \mathrm{Prob}\{\mathcal{W}_i = \mathrm{ACK}|\mathcal{V}_i = \mathrm{NACK}\}$ with $0 \leq P_\mathrm{A}, P_\mathrm{N} \leq 1/2$ in which $\mathcal{V}_i$ and $\mathcal{W}_i$ signify random processes.[1] The first error probability denotes the probability that an ACK is transmitted after the i-th data transmission ($1 \leq i \leq K$) but received as a NACK, while the latter probability indicates the opposite error scenario.[2] It is worth noting that P_A and P_N can be different in general. K is

[1] For $P_\mathrm{A} = P_\mathrm{N}$ this model is equivalent to a binary symmetric channel.

[2] P_A and P_N are either predefined according to given target error rates (e.g. 3GPP recommendations) or chosen arbitrarily in order to determine those target error rates under the

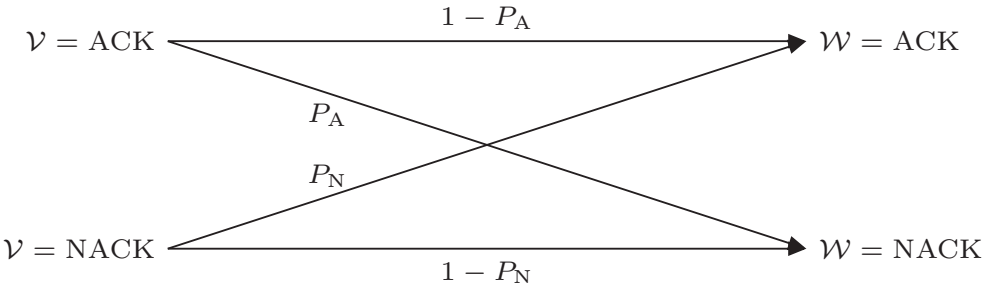


Figure 6.5: General model of the HARQ feedback channel.

the maximum number of transmissions allowed by the system. It is assumed, for the sake of simplicity, that the channel condition of the feedback channel remains constant during these K transmissions, yielding error probabilities P_A and P_N which are constant, i.e., independent of i. However, the resulting expressions can be adapted easily for fast varying block fading channels by substituting the fixed error probabilities P_A and P_N with probabilities P_A^i and P_N^i which can be different for each RV. The state probabilities after the i-th data transmission are given by means of the residual FERs P_i of the communication system according to $\mathrm{Prob}\{\mathcal{V}_i = \mathrm{NACK}\} = P_i$ and $\mathrm{Prob}\{\mathcal{V}_i = \mathrm{ACK}\} = 1 - P_i$. Assuming code combining at the receiver the following constraint holds true:

$$1 \geq P_1 \geq P_2 \geq \ldots \geq P_K \geq 0. \tag{6.11}$$

Depending on the considered communication channel, P_i can either be determined by simulations or even by analytical derivation.

A measure for the performance of a communication system is its average throughput per channel use which is defined for a system with reliable feedback by

$$\mathcal{G}_K^{[\mathrm{rel}]} = \frac{(1 - P_K) \cdot r \cdot I}{\overline{K}^{[\mathrm{rel}]}}. \tag{6.12}$$

It depends on the code rate r, the number I of coded bits per modulation symbol (e.g., OFDM subcarrier), the average number of transmissions $\overline{K}^{[\mathrm{rel}]}$ per frame and the residual error probability P_K after the K-th transmission. The factor $1 - P_K$ guarantees that, after the K-th transmission, only error-free decoded frames are considered for the calculation of $\mathcal{G}_K^{[\mathrm{rel}]}$. Equation (6.12) can be adopted to a system with unreliable feedback according to

$$\mathcal{G}_K^{[\mathrm{unrel}]} = C(K, P_\mathrm{N}, P_i) \frac{(1 - P_K) \cdot r \cdot I}{\overline{K}^{[\mathrm{unrel}]}} \tag{6.13}$$

$$= C(K, P_\mathrm{N}, P_i) \frac{\overline{K}^{[\mathrm{rel}]}}{\overline{K}^{[\mathrm{unrel}]}} \mathcal{G}_K^{[\mathrm{rel}]}, \tag{6.14}$$

where $C(K, P_\mathrm{N}, P_i)$ is a correction term which incorporates the fact that the NACK→ACK errors, in contrast to the ACK→NACK errors, increase the residual FER after HARQ. $\overline{K}^{[\mathrm{unrel}]}$ is the average number of transmissions per frame and depends on the error probabilities P_A, P_N, and P_i as well as the maximum number of transmissions K. Both terms will be derived for the considered system model in what follows. It is evident that $\mathcal{G}_K^{[\mathrm{unrel}]} = \mathcal{G}_K^{[\mathrm{rel}]}$ for $P_\mathrm{A} = P_\mathrm{N} = 0$.

condition of a maximum tolerated loss in throughput.

Average Number of Transmissions

The average number of transmissions per frame $\overline{K}^{[\text{unrel}]}$ is given by

$$\overline{K}^{[\text{unrel}]} = \sum_{i=1}^{K} i \cdot P_{i|K}, \tag{6.15}$$

where $P_{i|K} = \text{Prob}\{\mathcal{I} = i|\mathcal{K} = K\}$ denotes the probability that exactly i data transmissions are carried out under the condition that up to K transmissions are allowed by the system. $\mathcal{I}$ and $\mathcal{K}$ signify the corresponding random processes. For the probabilities $P_{i|K}$ it holds that

$$\sum_{i=1}^{K} P_{i|K} = 1. \tag{6.16}$$

These probabilities can be computed by means of the error probabilities P_{A}, P_{N}, P_i $(1 \leq i \leq K)$. For $K = 1$ the trivial solution is given by $P_{1|1} = 1$. For $K = 2$ there are 2 probabilities $P_{1|2}$ and $P_{2|2}$. Exactly 2 transmissions have to be carried out if either the first data transmission fails and the corresponding feedback does not fail or vice versa. In the rest of the cases, exactly 1 transmission has to be performed. Consequently, the resulting probabilities are given by

$$P_{1|2} = 1 - P_{2|2} \tag{6.17}$$

$$\begin{aligned} P_{2|2} &= \text{Prob}\{\mathcal{V}_1{=}\text{ACK}\} \cdot \text{Prob}\{\mathcal{W}_1{=}\text{NACK}|\mathcal{V}_1{=}\text{ACK}\} \\ &\quad + \text{Prob}\{\mathcal{V}_1{=}\text{NACK}\} \cdot \text{Prob}\{\mathcal{W}_1{=}\text{NACK}|\mathcal{V}_1{=}\text{NACK}\} \\ &= (1 - P_1)P_{\text{A}} + P_1(1 - P_{\text{N}}). \end{aligned} \tag{6.18}$$

If the number of allowed transmissions is increased to $K = 3$ the fraction of frames transmitted once does not change, i.e., $P_{1|3} = P_{1|2}$. With (6.16) the probability $P_{2|3}$ is given by $P_{2|3} = 1 - P_{1|3} - P_{3|3}$. Considering $P_{3|3}$, 3 scenarios are possible:

(1) Perfect decoding after the first transmission with probability $Q_1 = 1 - P_1$ and corrupted ACK feedback twice with probability P_{A}^2 results in the overall probability $(1 - P_1)P_{\text{A}}^2$ due to statistical independence.

(2) Faulty decoding after the second transmission with probability $Q_2 = P_2$ and error-free NACK feedback after the first and second transmission with probability $(1 - P_{\text{N}})^2$ results in the overall probability $P_2(1 - P_{\text{N}})^2$.

(3) For the remaining fraction of packets $Q_3 = 1 - Q_1 - Q_2 = 1 - (1 - P_1) - P_2 = P_1 - P_2$, three transmissions are requested for error-free NACK feedback after the first transmission and corrupted ACK feedback after the second transmission. This results in the overall probability $(P_1 - P_2)(1 - P_{\text{N}})P_{\text{A}}$.

Hence, the required probabilities for $K = 3$ transmissions can be summarized according to

$$P_{1|3} = P_{1|2} = 1 - P_{2|2} \tag{6.19}$$

$$P_{2|3} = 1 - P_{1|3} - P_{3|3} = P_{2|2} - P_{3|3} \tag{6.20}$$

$$P_{3|3} = (1 - P_1)P_{\mathrm{A}}^2 + P_2(1 - P_{\mathrm{N}})^2 + (P_1 - P_2)(1 - P_{\mathrm{N}})P_{\mathrm{A}}. \tag{6.21}$$

Figure 6.6 visualizes how the required probabilities are split up with increasing K. Based on the previous considerations three major properties of the given tree structure are highlighted in Figure 6.6 by gray boxes:

(a) $\displaystyle\sum_{i=1}^{K} P_{i|K} = 1 \quad \forall\, K$

(b) $P_{i|j} = P_{i|i+1} \quad \text{for} \quad i + 1 \leq j \leq K$

(c) $P_{i|i} = P_{i|i+1} + P_{i+1|i+1} \quad \text{for} \quad i \leq K - 1$

Property (c) can be proven with (a) and (b). For the example given above with $K = 3$ transmissions, the properties (b) and (c) can be observed in (6.19) and (6.20), respectively. Based on these properties, the equation for the average number

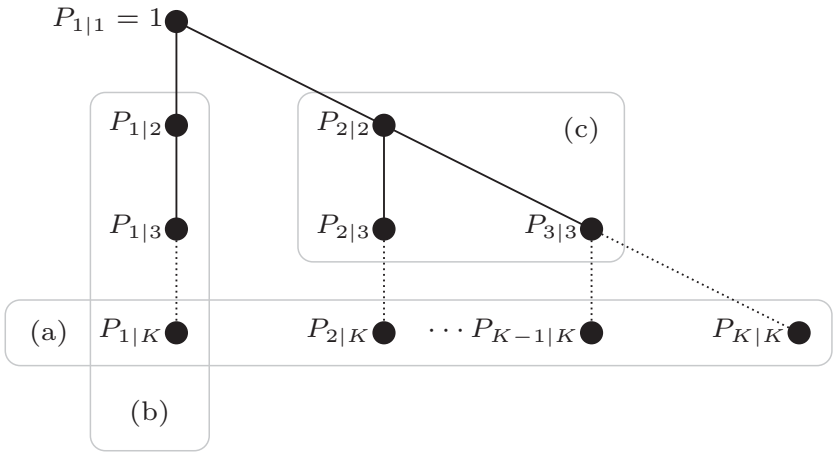


Figure 6.6: Tree-structure illustrating the computation rule for the average number of transmissions per frame.

of transmissions $\overline{K}^{[\mathrm{unrel}]}$, as given in (6.15), can be simplified according to

$$\begin{aligned}
\overline{K}^{[\mathrm{unrel}]} &= \sum_{i=1}^{K} i \cdot P_{i|K} \\
&= \sum_{i=1}^{K-1} i \cdot P_{i|K} + K P_{K|K} \\
&\overset{(b)}{=} \sum_{i=1}^{K-1} i \cdot P_{i|i+1} + K P_{K|K} \\
&\overset{(c)}{=} \sum_{i=1}^{K-1} i \left(P_{i|i} - P_{i+1|i+1} \right) + K P_{K|K} \\
&= \sum_{i=0}^{K-2} (i+1) P_{i+1|i+1} - \sum_{i=0}^{K-1} i \cdot P_{i+1|i+1} + K P_{K|K} \\
&= \sum_{i=0}^{K-2} P_{i+1|i+1} - (K-1) P_{K|K} + K P_{K|K} \\
&= \sum_{i=1}^{K-1} P_{i|i} + P_{K|K} \\
&= \sum_{i=1}^{K} P_{i|i}. \qquad (6.22)
\end{aligned}$$

As a result of (6.22), it is sufficient to compute the probabilities $P_{i|i}$ rather than all of the probabilities corresponding to the tree structure depicted in Figure 6.6. Accounting for the composition scheme of these probabilities revealed by (6.18) and (6.21), a direct computation rule can be formulated for any $2 \leq i \leq K$:

$$P_{i|i} = P_{i-1}(1 - P_{\mathrm{N}})^{i-1} + \sum_{j=1}^{i-1} (P_{i-j-1} - P_{i-j})(1 - P_{\mathrm{N}})^{i-j-1} P_{\mathrm{A}}^{j}, \qquad (6.23)$$

where $P_0 = 1$ for initialization.

System Throughput

For a communication system with $P_{\mathrm{N}} > 0$, the obtained throughput is reduced by a factor of $C(K, P_{\mathrm{N}}, P_i)$ which is obviously independent of the error probability P_{A}. Some corrupted frames are not retransmitted during the HARQ process since the corresponding NACK feedback is corrupted as well. The probability of that scenario can be determined dependent on K and the error probabilities P_i and P_{N}. Note that the last feedback message is sent after the $K-1$-th transmission:

(1) Probability of loosing a frame exactly after the first transmission: $P_1 P_\mathrm{N}$.

(2) Additional probability of loosing a frame exactly after the second transmission: $P_2 P_\mathrm{N}(1 - P_\mathrm{N})$.

$\vdots$

(i) Additional probability of loosing a frame exactly after the i-th transmission: $P_i P_\mathrm{N}(1 - P_\mathrm{N})^{i-1}$.

$\vdots$

(K-1) Additional probability of loosing a frame exactly after the second to last ($K-1$-th) transmission: $P_{K-1} P_\mathrm{N}(1 - P_\mathrm{N})^{K-2}$.

Based on the above considerations, it can be concluded that

$$C(K, P_\mathrm{N}, P_i) = 1 - P_\mathrm{N} \sum_{i=1}^{K-1} P_i \left(1 - P_\mathrm{N}\right)^{i-1} . \tag{6.24}$$

Hence, the overall throughput for a communication system with unreliable feedback channel is given by means of (6.13), (6.22), and (6.24) according to

$$\begin{aligned} \mathcal{G}_K^{[\mathrm{unrel}]} &= C(K, P_\mathrm{N}, P_i) \frac{(1 - P_K) \cdot r \cdot I}{\overline{K}^{[\mathrm{unrel}]}} \\ &= \left(1 - P_\mathrm{N} \sum_{i=1}^{K-1} P_i \left(1 - P_\mathrm{N}\right)^{i-1}\right) \frac{(1 - P_K) \cdot r \cdot I}{\sum_{i=1}^{K} P_{i|i}} \end{aligned} \tag{6.25}$$

with $P_{i|i}$ as derived in (6.23).

Link to Special Cases

Several special cases are covered by the expressions derived in the previous section, illustrating the general nature of the underlying feedback channel model.

1. Maximal unreliable feedback channel, i.e., $P_{\mathrm{A}} = P_{\mathrm{N}} = 1/2$:

$$\begin{aligned} P_{i|i} &= \left(\frac{1}{2}\right)^{i-1} \left(P_{i-1} + \sum_{j=1}^{i-1} (P_{i-j-1} - P_{i-j}) \right) \\ &= \left(\frac{1}{2}\right)^{i-1} \left(P_{i-1} + \sum_{j=1}^{i-1} P_{i-j-1} - \sum_{j=1}^{i-1} P_{i-j} \right) \\ &= \left(\frac{1}{2}\right)^{i-1} \left(P_{i-1} + \sum_{j=1}^{i-1} P_{i-j-1} - \sum_{j=0}^{i-2} P_{i-j-1} \right) \\ &= \left(\frac{1}{2}\right)^{i-1} (P_{i-1} + P_0 - P_{i-1}) = \left(\frac{1}{2}\right)^{i-1} \\ \overline{K}^{[\mathrm{unrel}]} &= \sum_{i=1}^{K} P_{i|i} = \sum_{i=1}^{K} \left(\frac{1}{2}\right)^{i-1} = \sum_{i=0}^{K-1} \left(\frac{1}{2}\right)^{i}. \end{aligned} \tag{6.26}$$

   This geometric sum can explicitly be computed according to

$$\overline{K}^{[\mathrm{unrel}]} = 2 \left(1 - \left(\frac{1}{2}\right)^{K} \right) = 2 - \left(\frac{1}{2}\right)^{K-1}. \tag{6.27}$$

   If the number of retransmission is infinite, i.e., $K \to \infty$, the average number of transmissions per frame results in

$$\overline{K}_{\infty}^{[\mathrm{unrel}]} = \lim_{K \to \infty} \left(2 - \left(\frac{1}{2}\right)^{K-1} \right) = 2. \tag{6.28}$$

   The overall throughput is finally determined by (6.25). It can be concluded that the maximal unreliable feedback channel has no impact on the overall system throughput for totally unreliable communication channels ($P_i = 1$ for $i < \infty$, i.e., $\mathcal{G}_{\infty}^{[\mathrm{unrel}]} = \mathcal{G}_{\infty}^{[\mathrm{rel}]} = 0$). However, assuming a perfect communication channel ($P_i = 0$ for $i < \infty$), the overall system throughput is reduced by the factor of two:

$$\mathcal{G}_{\infty}^{[\mathrm{unrel}]} = \frac{\overline{K}_{\infty}^{[\mathrm{rel}]}}{\overline{K}_{\infty}^{[\mathrm{unrel}]}} \mathcal{G}_{\infty}^{[\mathrm{rel}]} = \frac{1}{2} \mathcal{G}_{\infty}^{[\mathrm{rel}]}. \tag{6.29}$$

2. Reliable (perfect) feedback channel according to [SCV04], i.e., $P_{\mathrm{A}} = P_{\mathrm{N}} = 0$:

$$\begin{aligned} \overline{K}^{[\mathrm{unrel}]} = \overline{K}^{[\mathrm{rel}]} &= \sum_{i=1}^{K} P_{i|i} = 1 + \sum_{i=2}^{K} P_{i|i} \\ &= 1 + \sum_{i=2}^{K} P_{i-1} = 1 + \sum_{i=1}^{K-1} P_i \end{aligned} \tag{6.30}$$

$$C(K, P_{\mathrm{N}}, P_i) = 1. \tag{6.31}$$

The overall throughput is then given by

$$\mathcal{G}_K^{[\text{unrel}]} = \mathcal{G}_K^{[\text{rel}]} = \frac{(1 - P_K) \cdot r \cdot I}{1 + \sum_{i=1}^{K-1} P_i}, \tag{6.32}$$

allowing the calculation of the throughput only based on FER simulations.

3. Perfect communication channel, i.e., $P_i = 0$ for $1 \leq i \leq K$, and feedback channel with $0 \leq P_\text{A} \leq 1/2$ and $0 \leq P_\text{N} \leq 1/2$:

$$\begin{aligned} \overline{K}^{[\text{unrel}]} &= 1 + \sum_{i=2}^{K} P_{i|i} = 1 + \sum_{i=2}^{K} P_\text{A}^{i-1} \\ &= 1 + \sum_{i=1}^{K-1} P_\text{A}^{i} = \sum_{i=0}^{K-1} P_\text{A}^{i} = \frac{1 - P_\text{A}^K}{1 - P_\text{A}} \end{aligned} \tag{6.33}$$

$$C(K, P_\text{N}, P_i) = 1. \tag{6.34}$$

For such a system, (6.25) simplifies to

$$\mathcal{G}_K^{[\text{unrel}]} = \frac{(1 - P_\text{A}) \cdot r \cdot I}{1 - P_\text{A}^K}. \tag{6.35}$$

4. Erasure feedback channel:
In many previous works, e.g., [ML00, AN07, Bad09], the feedback channel is modeled as an erasure channel which is also covered by the general framework presented in this section. It is often assumed that the ACK/NACK feedback is protected by a CRC. If an erasure occurs, this information is always interpreted as a NACK which secures a cross over probability of $P_\text{N} = 0$. Applied to the general expressions given in (6.22), (6.23) and (6.25), this

results in an average number of transmissions per frame of

$$\begin{aligned}
\overline{K}^{[\mathrm{unrel}]} &= \sum_{i=1}^{K}\left(\sum_{j=1}^{i-1}(P_{i-j-1}-P_{i-j})P_{\mathrm{A}}^{j}+P_{i-1}\right)\\
&= 1+\sum_{i=1}^{K-1}P_i+\sum_{i=2}^{K}\sum_{j=1}^{i-1}(P_{i-j-1}-P_{i-j})P_{\mathrm{A}}^{j}\\
&= 1+\sum_{i=1}^{K-1}P_i+\sum_{i=1}^{K-1}\sum_{j=1}^{i}(P_{i-j}-P_{i-j+1})P_{\mathrm{A}}^{j}\\
&= 1+\sum_{i=1}^{K-1}P_i+\sum_{i=1}^{K-1}\sum_{j=1}^{i}P_{i-j}P_{\mathrm{A}}^{j}-\sum_{i=1}^{K-1}\sum_{j=1}^{i}P_{i-j+1}P_{\mathrm{A}}^{j}\\
&= 1+\sum_{i=1}^{K-1}P_i+\sum_{i=0}^{K-2}\sum_{j=1}^{i+1}P_{i-j+1}P_{\mathrm{A}}^{j}-\sum_{i=1}^{K-1}\sum_{j=1}^{i}P_{i-j+1}P_{\mathrm{A}}^{j}\\
&= 1+\sum_{i=1}^{K-1}P_i+P_0P_{\mathrm{A}}-\sum_{j=1}^{K-1}P_{K-j}P_{\mathrm{A}}^{j}+\sum_{i=1}^{K-2}P_0P_{\mathrm{A}}^{i+1}\\
&= 1+\sum_{i=1}^{K-1}P_i+\sum_{i=1}^{K-1}P_{\mathrm{A}}^{i}-\sum_{i=K-1}^{1}P_iP_{\mathrm{A}}^{K-i}\\
&= 1+\sum_{i=1}^{K-1}P_i+\sum_{i=1}^{K-1}\left(P_{\mathrm{A}}^{i}-P_iP_{\mathrm{A}}^{K-i}\right) \qquad (6.36)
\end{aligned}$$

which is identical to the expression derived in [ML00].

Analytical Solution vs. Simulation Results

In order to evaluate the theoretical throughput of a communication system with unreliable feedback as given in (6.25), LTE as an exemplary physical layer has been simulated using an AWGN channel as well as the specified 3GPP TU 20MHz channel modeling a typical urban environment. Simulations have been conducted with 10 Turbo iterations, a maximum of $K = 4$ transmissions, and with modulation and coding schemes as indicated in Table 6.1. The residual FERs have been simulated with an error-free feedback channel ($P_{\mathrm{N}} = P_{\mathrm{A}} = 0$). Based on these results, the average number of transmissions and the system throughput can be calculated analytically with (6.22) and (6.25), respectively, for any feedback error probabilities P_{A} and P_{N}. The analytical results have been verified by system simulations for $P_{\mathrm{N}} = P_{\mathrm{A}} \in \{0.1, 0.2\}$.

Figure 6.7 shows the average number of transmissions of the LTE system, exemplarily for 16QAM ($I = 4$) with a code rate of $r_{\mathrm{RM}} = 4/5$. It can be seen

that the number of necessary transmissions obviously decreases for an increasing channel quality. Additionally, a higher feedback error probability ($P_{\mathrm{A}} = P_{\mathrm{N}} > 0$) results in less average transmissions for low channel qualities. For high channel qualities ($P_i \to 0$ for $1 \leq i \leq 4$), the average number of transmission converges corresponding to (6.33). Most importantly, it can be observed that the simulations match the analytical results perfectly, thus, confirming the validity of the derivation.

In Figure 6.8, the envelope of the throughput curves for all modulation and coding schemes is depicted. In case of unreliable feedback ($P_{\mathrm{A}} = P_{\mathrm{N}} > 0$), the curve progression is similar to error-free feedback, but the overall throughput decreases for higher feedback error probabilities. The perfect match of the simulated and the calculated throughput support the validity of the analytical evaluation as well. It is obvious that the simulation of a system with unreliable feedback channel is therefore superfluous, since the throughput can just be calculated analytically for any feedback error probabilities only based on the simulation of the system with reliable feedback.

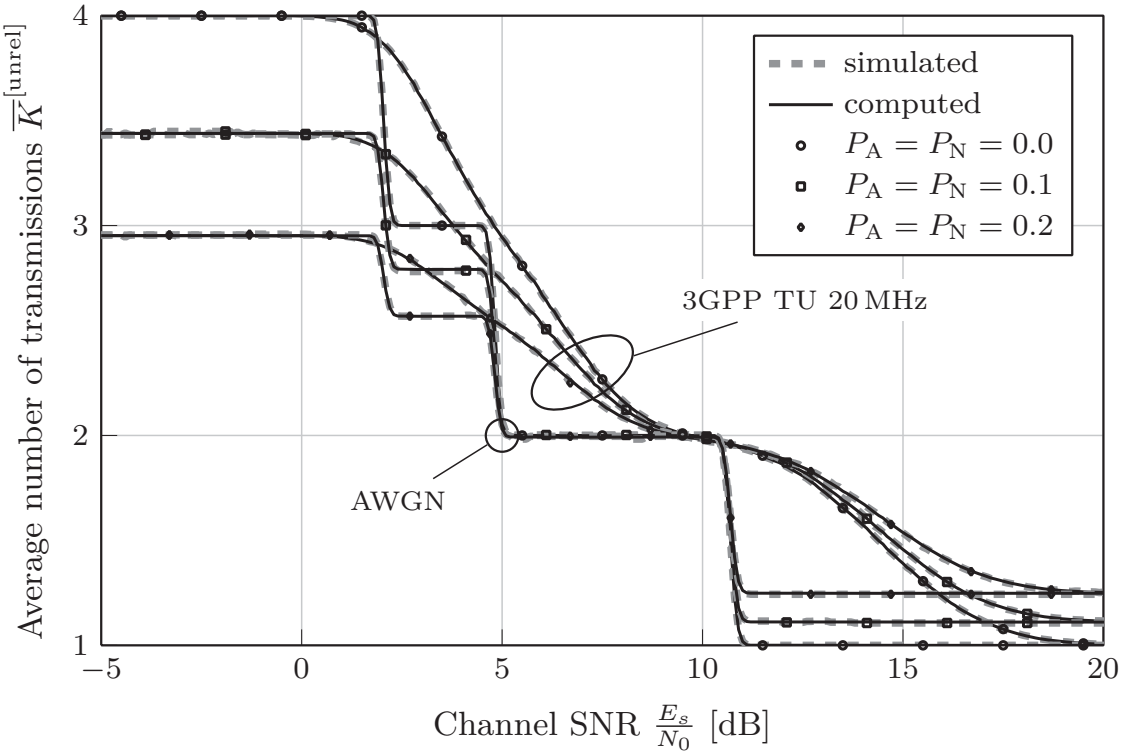


Figure 6.7: Average number of transmissions for $I = 4$ and $r_{\mathrm{RM}} = 4/5$.

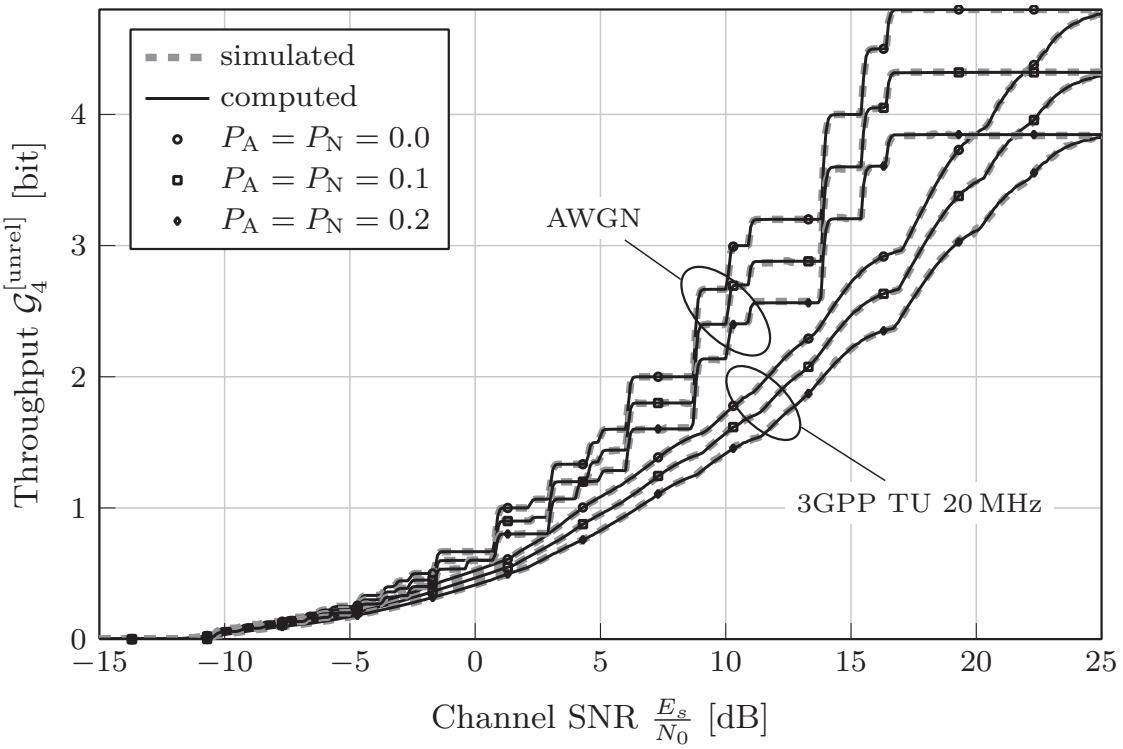


Figure 6.8: Maximum throughput (envelope) for all modulation and coding schemes.

Chapter 7

Header Bit Assisted Channel Decoding

The principle of *Header bit Assisted Channel Decoding* (HACD) has initially be presented in [BLV+10] for a system based on *Iterative Source-Channel Decoding* (ISCD) [Gör00, AVS01]. ISCD is a turbo-like decoding scheme in which a source decoder, employing the *Soft Decision Source Decoding* (SDSD) [FV01] principle, and a convolutional *Soft-In/Soft-Out* (SISO) decoder iteratively exchange *extrinsic* information in order to improve jointly the overall decoding performance.

However, in conventional packet-switched transmission systems, the channel decoder is located at the physical layer and the source decoder at the application layer. Therefore, there is no inherent possibility for exchanging *extrinsic* soft information and ISCD can only be carried out if the complete packet header is correctly decoded, which guarantees a successful delivery of the packet.

Besides the feedback of *extrinsic* information computed by the SDSD decoder, perfect *extrinsic* information about the known header bits can additionally be fed back to the channel decoder. These bits take the role of dummy bits within the decoding process and assist the decoding of the packet payload, resulting in an increased overall system throughput. So far, protocol headers have only been seen as throughput limiting factor that are required for packet routing rather than being a support for iterative decoding schemes.

It is worth mentioning that the application of HACD is not limited to ISCD and can be adopted to systems employing convolutional Turbo codes or LDPC codes as well. A general model for a transmission system using HACD is illustrated in Figure 7.1. The payload $\mathbf{v}$ of length ℓ_v is given as input. Furthermore, a header $\breve{\mathbf{h}}$ of length $\ell_{\breve{h}}$ is composed by the intermediate network layers and may be protected (optional) against transmission errors using an appropriate *Forward Error Correction* (FEC) code. The resulting vector $\mathbf{h}$ of length ℓ_h is then attached to the payload $\mathbf{v}$ forming the packet $\mathbf{b} = [\mathbf{h}, \mathbf{v}]$ of length $\ell_b = \ell_v + \ell_h$. Identical to systems using dummy bit insertion, HACD requires an interleaver π which spreads the header bits over the complete packet. This interleaver follows the design guidelines developed for *Rate-Compatible Insertion Convolutional* (RCIC) (Turbo) codes (see Section 3.3) or LDPC codes using dummy bit insertion (see Section 5.2). While in the former case, the header bits should be equidistantly distributed within the packet, in the latter case, an appropriate distribution is

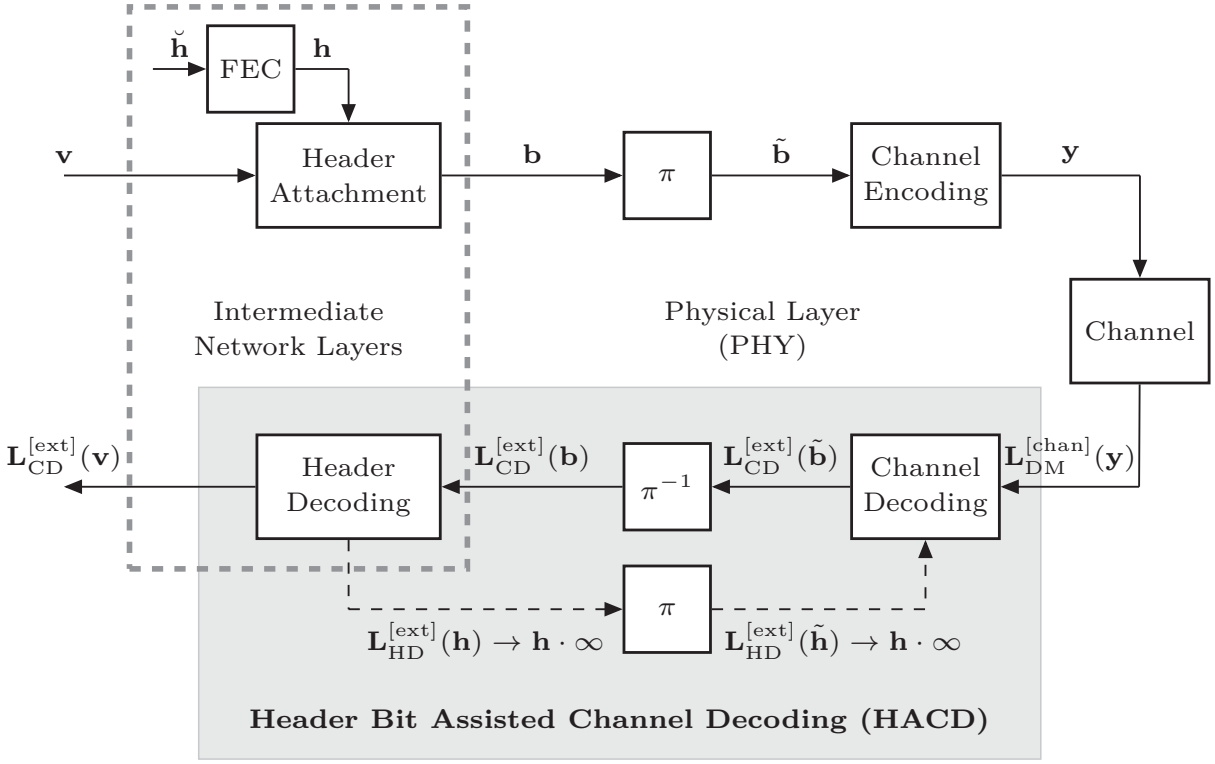


Figure 7.1: General transmission system model using HACD.

achieved by using Algorithm 5.1, in which those positions (variable nodes) are selected for header bits which have approximately equal distance within the tanner graph. The interleaved packet $\tilde{\mathbf{b}}$ is then encoded by the channel encoder (including rate matching) of code rate r_{RM} yielding the output packet $\mathbf{y}$ of length $\ell_y = r_{\mathrm{RM}}^{-1} \cdot \ell_b$ which is transmitted over the channel.

At the receiver, channel-related *Log-Likelihood Ratio*s (LLRs) $\mathbf{L}_{\mathrm{DM}}^{[\mathrm{chan}]}(\mathbf{y})$ are exploited by the SISO channel decoder to generate *extrinsic* (reliability) information $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{ext}]}(\tilde{\mathbf{b}})$ about the interleaved packet $\tilde{\mathbf{b}}$. After deinterleaving, header decoding is performed based on the *extrinsic* information $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{ext}]}(\mathbf{h})$ of the FEC encoded header $\mathbf{h}$. If an erroneous header is detected by the included checksums or a separate *Cyclic Redundancy Check* (CRC), the complete packet is discarded. Otherwise, perfect *extrinsic* information $\mathbf{L}_{\mathrm{HD}}^{[\mathrm{ext}]}(\mathbf{h}) \rightarrow \mathbf{h} \cdot \infty$ of $\mathbf{h}$ is fed back to the channel decoder supporting the second decoding process. The improved reliability values $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{ext}]}(\mathbf{v})$ of the payload $\mathbf{v}$ are finally delivered to the target application which can either start a further iterative loop (e.g. ISCD) or compute the final estimate of the payload. Erroneous payloads can then be used, e.g, for an application-specific concealment [BV10a].

However, HACD can only be carried out if a number of network stack modifications are introduced. These network architecture aspects are discussed in Section 7.1.

Furthermore, the effectiveness of HACD is highly dependent on the success of the header decoding unit. Therefore, novel principles as *Soft Decision Header*

Decoding (SDHD) also known as *Joint Protocol-Channel Decoding* (JPCD) [DK10, AKD12,KD10] should be employed. SDHD[1] exploits the reliability values $\mathbf{L}_{\mathrm{CD}}^{[\mathrm{ext}]}(\mathbf{h})$, the information given by the FEC, as well as the *a priori* knowledge provided by the network stack itself to perform an improved estimation of the header content. A brief overview of all types of usable network redundancies is given in Section 7.2.

However, it is not in the scope of this chapter to develop the optimal configuration of HACD-based systems, but to provide a bound on the performance of HACD depending on the packet header fraction. If Turbo coding is employed, such a bound can be derived using the expressions on EXIT charts for RCIC Turbo codes derived in Section 4.3 as it will be shown in Section 7.3.

7.1 Network Architecture Aspects

The architecture of conventional network stacks is organized in accordance to the so-called TCP/IP model which is related to the ISO-OSI reference model [ISO96] and subdivided into four logical layers, namely the *application layer*, the *transport layer*, the *internet layer*, and the *link layer*. Interfaces exist only between adjacent layers (*vertical communication*), i.e., a joint processing across an arbitrary number of layers is strictly prohibited by conventional networks. Furthermore, all layers support a distinct set of protocols defining rules for a layer-specific communication between the transmitter and the receiver (*horizontal communication*). Obviously, the selected protocols depend on the signaling requirements of the considered application. In delay sensitive applications as, e.g., in real-time voice, audio, or video communication as well as streaming applications, signaling is commonly carried out by means of the *Real-time Transport Protocol* (RTP) [RFC96], the *User Datagram Protocol* (UDP) [RFC80], the *Internet Protocol* (IP) version 4 or 6 [RFC81,RFC98], as well as architecture-specific link layer protocols.

As an example, the user plane protocol stack specified in the LTE system [FYCL11] is depicted in Figure 7.2. Besides the architecture-independent protocol stack (RTP/UDP/IP), there are four additional protocol layers defined (marked in light gray): The *Packet Data Convergence Protocol* (PDCP) layer processes the IP packets delivered by the IP layer and is mainly responsible for header compression (if supported by both, the transmitted and the receiver) and security issues. Main functions of the *Radio Link Control* (RLC) layer are frame size adaptations by segmentation and reassembly of upper layer packets, as well as packet reordering caused by out-of-order reception during *Hybrid Automatic Repeat reQuest* (HARQ). The *Medium Access Control* (MAC) layer multiplexes data received from different radio bearers and attempts to achieve the negotiated *Quality of Service* (QoS). Furthermore, it is responsible for the control of the HARQ process. As illustrated in the previous chapters, the *Physical Layer* (PHY) comprises error

[1] Throughout this chapter, the term SDHD is used due to the strong relation to the SDSD principle proposed by Fingscheidt in 2001.

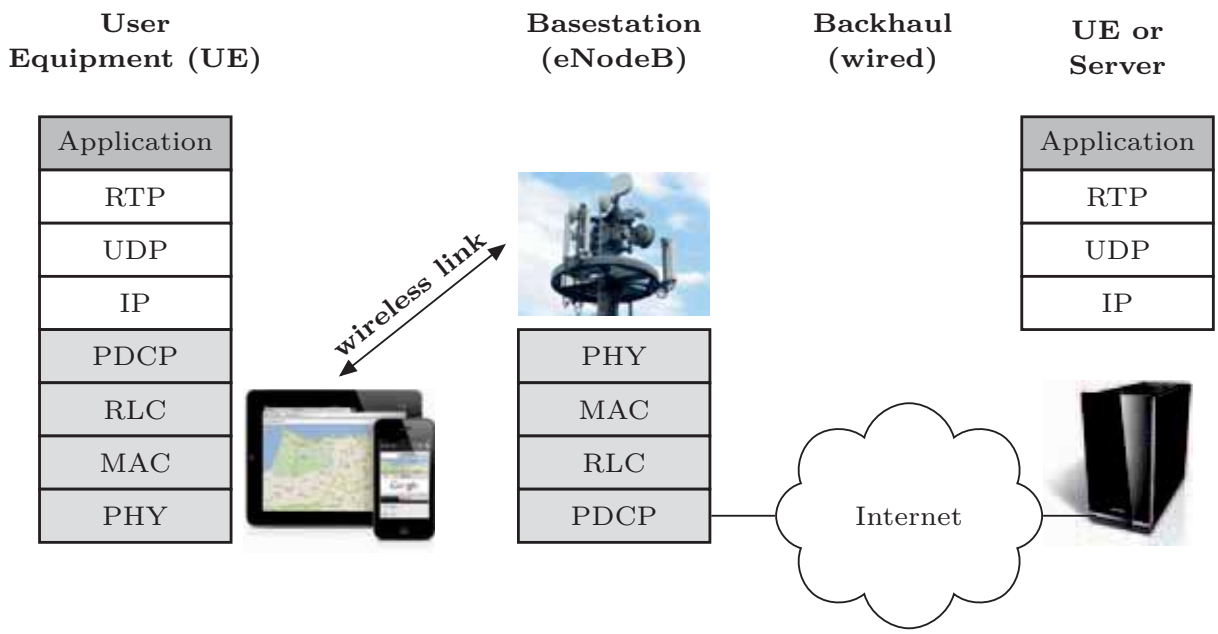


Figure 7.2: User plane protocol stack specified in LTE.

protection capabilities.

While the complete network stack is implemented in the mobile device, only link layer specific functions are provided by the base station. The remaining protocol stack is spatially separated and assigned to the far-end, e.g., a server providing multimedia content within a streaming session. In general, HACD delivers feedback information about each of the implemented network layers. However, it is evident that a feedback through the complete backhaul network is impossible due to delay constraints and the significantly increased network traffic demand. Considering the uplink scenario (transmission from the mobile device to the base station), HACD is only possible for the basestation-related layers. In the downlink (opposite transmission direction), however, HACD can be carried out across the complete network supporting the channel decoding process by a considerably higher fraction of known bits.

Moreover, to enable HACD in packet-switched networks, a number of network stack modifications are assumed:

- The considered network stack is capable of processing soft information required for SDHD and HACD.
- In current IP-based transmission systems, a physical layer CRC detects erroneous packets after channel decoding. These packets are directly discarded resulting in a considerable loss in spectral efficiency. While packet headers need to be absolutely error-free, erroneous payloads may still be utilized by the source decoder. Therefore, the consistency check in the considered network stack only covers the protocol headers, while the handling of erroneous payloads is assigned to the source decoder. This can be realized, e.g., by an

additional CRC at the MAC layer.

- In contrast to conventional network stacks, the considered stack enables cross-layer communication, i.e., the communication between non-adjacent layers. This is required in particular for the transmission of header information back to the physical layer, but also for the efficient utilization of the complete network redundancy (see Section 7.2).

The flexible cross-layer architecture *CRAWLER*, proposed in [AOSW10], builds a suitable base, since it enables signaling between an arbitrary number of layers and system components avoiding severe intrusions into the network stack. Moreover, it provides a high potential framework to adapt the architecture to the above mentioned requirements. These two properties are the main advantages compared to other cross-layer architectures proposed in literature [CMTG04, RI04].

7.2 Soft Decision Header Decoding

The *Soft Decision Header Decoding* (SDHD) principle is based on the permeable protocol layer mechanism introduced in [JSX05] and enhanced in a number of further contributions [MLKD08a, MLKD08b, KD10]. This mechanism includes the optimal exploitation of the redundancy given within the network layers in order to warrant a robust decoding of the header content. Furthermore, most of the contributions additionally uses the redundancy provided by the MAC layer CRC. A detailed study of this concept is presented in [MLKD10] and an enhancement to upper layers (IP/UDP) using checksums is described in [MK10]. However, although this will provide substantial *a priori* knowledge to the SDHD decoder, disregarding the consistency checks might lead to severe packet routing errors. The utilization of this concept should, therefore, be discussed for each application scenario individually.

In general, the SDHD decoder exploits two different types of network redundancy: (1) *time-invariant* redundancy inherently given by the protocols, i.e., *protocol-related* redundancy, as well as (2) *time-variant* redundancy which is provided by the currently available flows within the network stack, i.e., *flow-related* redundancy. These types of redundancy enable the classification of the protocol header fields into four classes: *known*, *predictable*, *unknown with a priori information*, and *unknown without a priori information*.

The *version field* of the IPv4 header, e.g., is always 4 and, thus, *known*.

An example for an *predictable* field is the *length field* of the IPv4 header, which covers the length of the IPv4 header $\ell^{\mathrm{H}}_{\mathrm{IPv4}}$ and the IPv4 payload $\ell^{\mathrm{P}}_{\mathrm{IPv4}}$. The overall length $\ell^{\mathrm{H+P}}_{\mathrm{IPv4}}$ is then predictable based on the known length $\ell^{\mathrm{H}}_{\mathrm{Link}}$ of the Link layer headers, which have already been retrieved, and the length ℓ_{b} of the entire packet $\mathbf{b}$ including the payload and all headers:

$$\ell^{\mathrm{H+P}}_{\mathrm{IPv4}} = \ell^{\mathrm{H}}_{\mathrm{IPv4}} + \ell^{\mathrm{P}}_{\mathrm{IPv4}} = \ell_{\mathrm{b}} - \ell^{\mathrm{H}}_{\mathrm{Link}}. \tag{7.1}$$

Consequently, transmission errors within predictable fields can be corrected without residual errors.

An *unknown field with a priori information* is, e.g., the *destination port filed* of the UDP header. The destination port (16 bits) is one of the open ports at the receiver. Since it is unlikely that $2^{16} = 65536$ ports are open at the same time, this field provides much *a priori* information to the SDHD decoder.

An example for an *unknown filed without a priori information* is the *hop limit field* in the IPv6 header. This field is initialized with a certain value and decremented whenever a hop occurs (passing a gateway). The initial value as well as the number of hops is unknown at the receiver resulting in a field without any *a priori* information. A complete classification of the RTP/UDP/IP protocol stack is carried out in Appendix D. Session set-up signaling (SIP/SDP) has been disregarded in this evaluation, i.e., that each incoming packet is assigned to exactly one currently active flow.

Finally, MAP estimation is employed by the SDHD decoder to recover the unknown header fields [MLKD10]. The expressions presented in [ACS08] (MAP SDSD) can be used for the computation of the required *a posteriori* probabilities. An exact derivation of the expressions is omitted here. The reader is referred to [ACS08, AVS01, Gör00] for a detailed study on this topic.

Further related work

The SDHD principle described above has also been applied to related topics in wireless or mobile communication as, e.g., robust frame synchronization [AKD09, AKD10, AKD12]. Frame synchronization is carried out at the receiver to recover single frames within a burst of aggregated frames. A detailed survey of all concepts described above is given in [DK10]. The header recovery procedure *Refector* proposed in [SAAW11] also exploits the redundancy inherently given by the network stack. Furthermore, Refector performs a classification of the header fields in terms of their importance for a successful packet delivery. This establishes an *error-tolerant* network stack, since packets with erroneous header field can be delivered to the correct target application. However, only perfect *extrinsic* information about the error-free decoded header fields can be fed back to the channel decoder.

7.3 Performance Bound on HACD

The gains provided by HACD are highly dependent on the employed header insertion scheme, the fraction of inserted header bits, and the effectiveness of the header decoding unit.

Assuming randomly distributed header bits, a bound on the system performance can be derived based on the semi-analytical EXIT chart analysis described in Section 4.3.2 for rate matching by means of dummy bit insertion. This bound is exactly reached if all headers have correctly been retrieved after each trans-

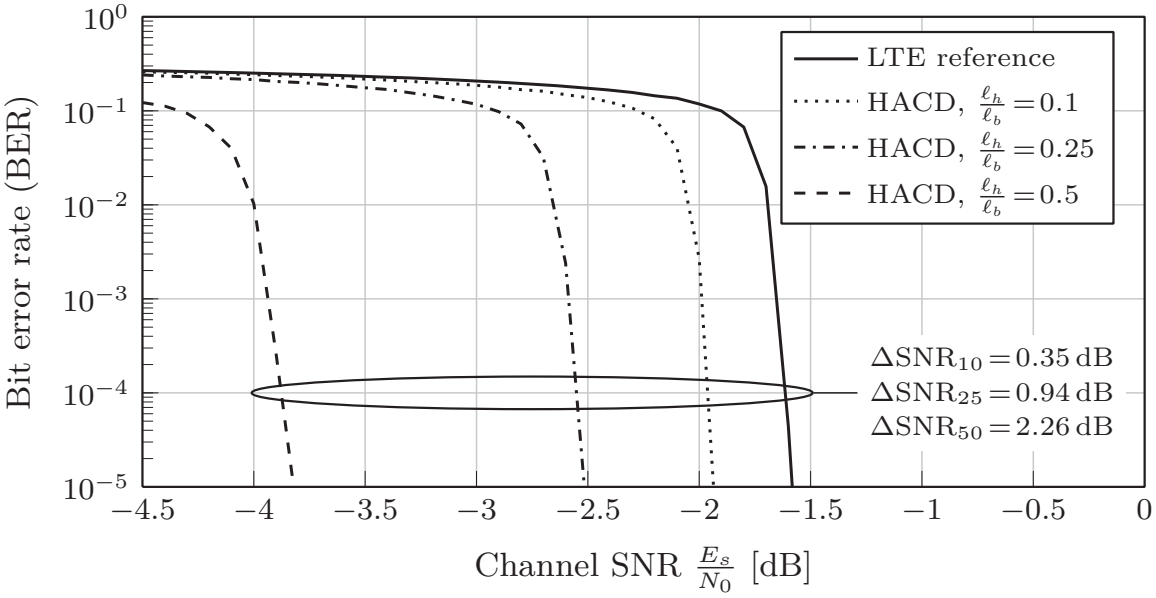


Figure 7.3: Performance Bounds on the BER for HACD and different header fraction.

mission. Usually, extensive BER or FER simulations have to be carried out to determine the respective performance bounds. Such a BER simulation is depicted in Figure 7.3 for the rate-1/3 LTE mother code with QPSK modulation. A header fraction of $\ell_h/\ell_b \in \{10\,\%, 25\,\%, 50\,\%\}$ has been randomly distributed within the packet of length $\ell_b = 100000\,\mathrm{bits}$[2]. Furthermore, 9 Turbo iterations have been carried out by the Turbo decoder after header decoding. As mentioned above, perfect header decoding has been assumed. Considering Figure 7.3 at a target *Bit Error Rate* (BER) of 10^{-4} , the bounds show considerable maximum performance improvements in terms of channel *Signal-to-Noise Ratio* (SNR) compared to the LTE reference (no HACD). For the evaluated scenarios, they amount to $\Delta\mathrm{SNR}_{10} = 0.35\,\mathrm{dB}$, $\Delta\mathrm{SNR}_{25} = 0.94\,\mathrm{dB}$, and $\Delta\mathrm{SNR}_{50} = 2.26\,\mathrm{dB}$, where the index denotes the header fraction.

Similar results have been obtained by the semi-analytical EXIT chart analysis. Known from the BER simulation of the LTE reference system, 9 Turbo iterations are sufficient for near-perfect decoding at a channel SNR of $E_s/N_0 = -1.6\,\mathrm{dB}$. This has also been observed by the corresponding EXIT chart illustrated in Figure 7.4 (left). In this case, the openness of the decoding tunnel is given by $\Omega = 0.03$. For worse channel conditions, Algorithm 4.3 (see Section 4.3.2) can be used to compute the fraction of known header bits ℓ_h/ℓ_b, which are required to hold $\Omega = 0.03$. The result of this analysis is shown in Figure 7.4 (right). The gain $\Delta\mathrm{SNR} =$

[2]It is evident that for large packet lengths, the header fraction is much less than 50 % in real scenarios. However, the evaluation by means of EXIT charts inherently assumes packet lengths of infinity. Therefore, large packet lengths have initially been assumed to show the accuracy of the presented analysis. In further evaluations this assumption is abandoned.

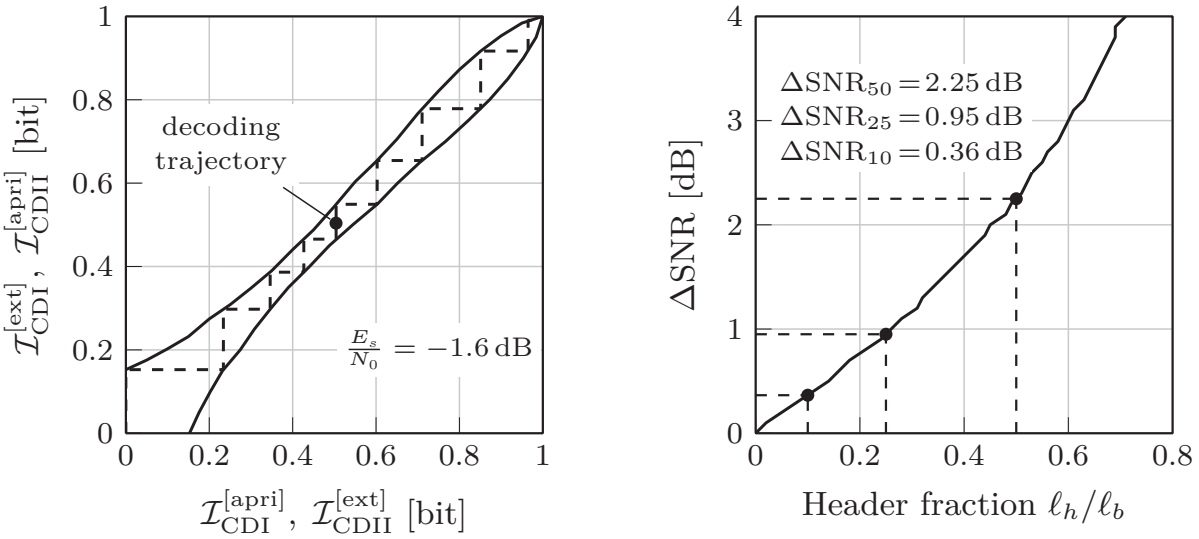


Figure 7.4: EXIT chart of the LTE mother code for QPSK modulation (left) and performance bound in terms of ΔSNR dependent on the header fraction ℓ_h/ℓ_b (right).

Modulation	Code rates
QPSK ($I = 2$)	$r_{\mathrm{RM}} \in \{\frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}\}$
16QAM ($I = 4$)	$r_{\mathrm{RM}} \in \{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}\}$
64QAM ($I = 6$)	$r_{\mathrm{RM}} \in \{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}\}$

Table 7.1: Modulation and coding schemes (MCS).

SNR − 1.6 dB is depicted in dependence of the inserted header fraction ℓ_h/ℓ_b. The specific header fractions $\ell_h/\ell_b \in \{10\,\%, 25\,\%, 50\,\%\}$, which have been evaluated in Figure 7.3, are marked by black dots. Accordingly, the bounds determined by the semi-analytical EXIT chart analysis closely match the bounds obtained from extensive BER simulations. Only EXIT chart simulations of the LTE reference system at the specific channel SNRs have to be carried out.

However, the given bounds are only valid for HACD-systems using random header bit insertion. For deterministically inserted header bits, which are, e.g., equidistantly distributed within the packet by means of Algorithm 3.1, such a bound cannot be derived theoretically and extensive simulations have to be carried out instead. Nevertheless, the semi-analytical results provide close approximations on the actual bounds as it is shown in the following example.

Evaluation of LTE employing HACD

In a first experiment, the LTE system has been simulated in an AWGN environment for the *Modulation and Coding Schemes* (MCSs) given in Table 7.1. Furthermore, the transmitted packets of length $\ell_b = 6144$ include a header fraction ℓ_h/ℓ_b of

either 10 %, 25 %, or 50 % which has been equidistantly distributed within the packet according to Algorithm 3.1. A maximum of $K = 4$ transmissions per packet are allowed by the system. The performance has been measured for each MCS in terms of system throughput which is defined for packet-switched communication systems according to

$$\mathcal{G}_{\mathrm{mcs}} = (1 - \mathrm{FER}) \cdot \frac{r_{\mathrm{RM}} \cdot I}{\overline{K}} \cdot \frac{\ell_v}{\ell_b}, \tag{7.2}$$

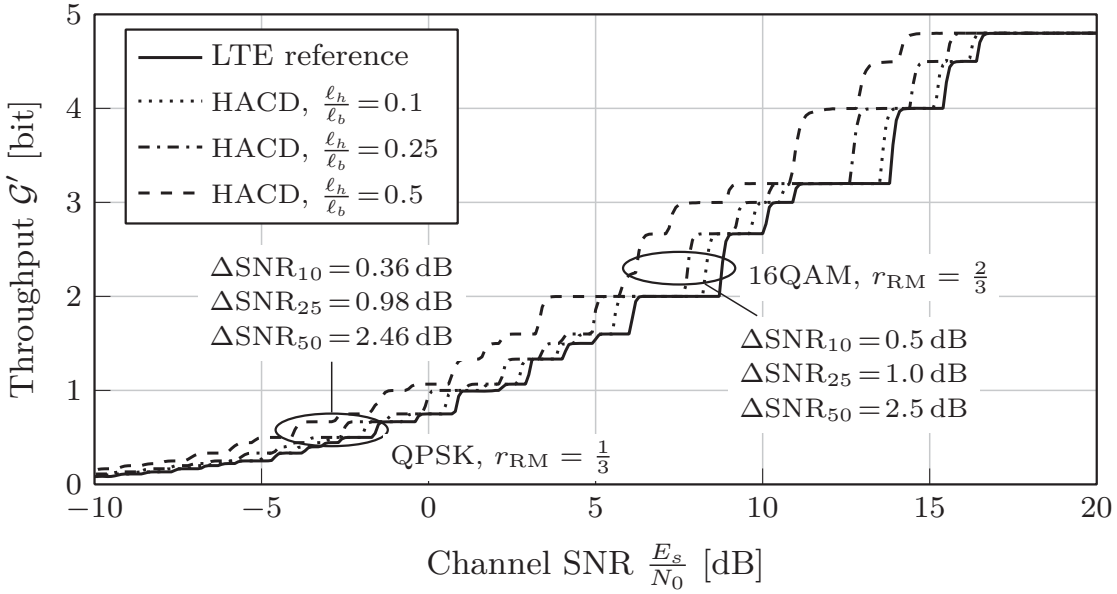
in which FER denotes the residual frame error rate after HARQ, I the number of bits per modulation symbol, and $\overline{K}$ the average number of transmissions per packet. However, for the determination of the gains achieved by HACD in terms of channel SNR, it is sufficient to compare the scaled versions $\mathcal{G}'_{\mathrm{mcs}} = \ell_b/\ell_v \cdot \mathcal{G}_{\mathrm{mcs}}$ of the actual throughput curves $\mathcal{G}_{\mathrm{mcs}}$. The reason is that the scaling factor effects only the value on the throughput axis, while the value on the channel SNR axis remains unchanged. To this end, the envelope $\mathcal{G}'$ of all curves $\mathcal{G}'_{\mathrm{mcs}}$ can be plotted in a single figure for different header fractions as shown in Figure 7.5(a). All HACD throughput curves have then the identical reference curve (solid line).

The bounds for QPSK and rate-1/3 Turbo coding are slightly higher than the bounds predicted by the EXIT chart analysis (see Figures 7.4 and 7.5), since the header bits are inserted at optimized positions. Nevertheless, the semi-analytically derived performance bounds for random insertion provide very close approximations in particular for moderate header fractions $\ell_h/\ell_b < 25\,\%$. It has further been observed that the gains depend on the employed modulation and coding scheme. For example, gains up to $\Delta\mathrm{SNR}_{10} = 0.5\,\mathrm{dB}$, $\Delta\mathrm{SNR}_{25} = 1.0\,\mathrm{dB}$, and $\Delta\mathrm{SNR}_{50} = 2.5\,\mathrm{dB}$ can be achieved by HACD for 16QAM and rate-2/3 Turbo coding.

In addition to the AWGN simulation, the Typical Urban (TU) channel model specified by 3GPP for a carrier bandwidth of 20 MHz has been used (see Appendix A). The scaled throughput $\mathcal{G}'$ of the LTE system employing HACD is depicted in Figure 7.5(b) for the considered scenarios $\ell_h/\ell_b \in \{10\,\%, 25\,\%, 50\,\%\}$. Again, it has been assumed that all headers have been retrieved without errors during the header decoding process. Therefore, the revealed gains in terms of channel SNR have to be interpreted as bounds on the actual system performance. Nevertheless, considering Figure 7.5(b), gains up to $\Delta\mathrm{SNR}_{10} = 0.7\,\mathrm{dB}$, $\Delta\mathrm{SNR}_{25} = 1.3\,\mathrm{dB}$, and $\Delta\mathrm{SNR}_{50} = 3.6\,\mathrm{dB}$ are provided by HACD in this simulation scenario.

As expected, the gains highly depend on the header fraction of the packet. Ericsson Research has presented in [Per07] system evaluations for *Voice over IP* (VoIP) over LTE. In order to reduce the packet overhead, *Robust Header Compression* (RoHC) has been employed which substitutes the original 40 byte RTP/UDP/IPv4 packet header by a 3 byte compressed version. In addition to the RoHC header, a PDCP header of 2 bytes and a RLC/MAC header of 5 bytes have been attached to the 35 byte VoIP payload. This results in a considerable overhead of $\ell_h/\ell_b \approx 22\,\%$ per packet. However, RoHC dramatically reduces the amount of redundancy given

(a) Transmission channel model: AWGN



(b) Transmission channel model: 3GPP TU@20 MHz

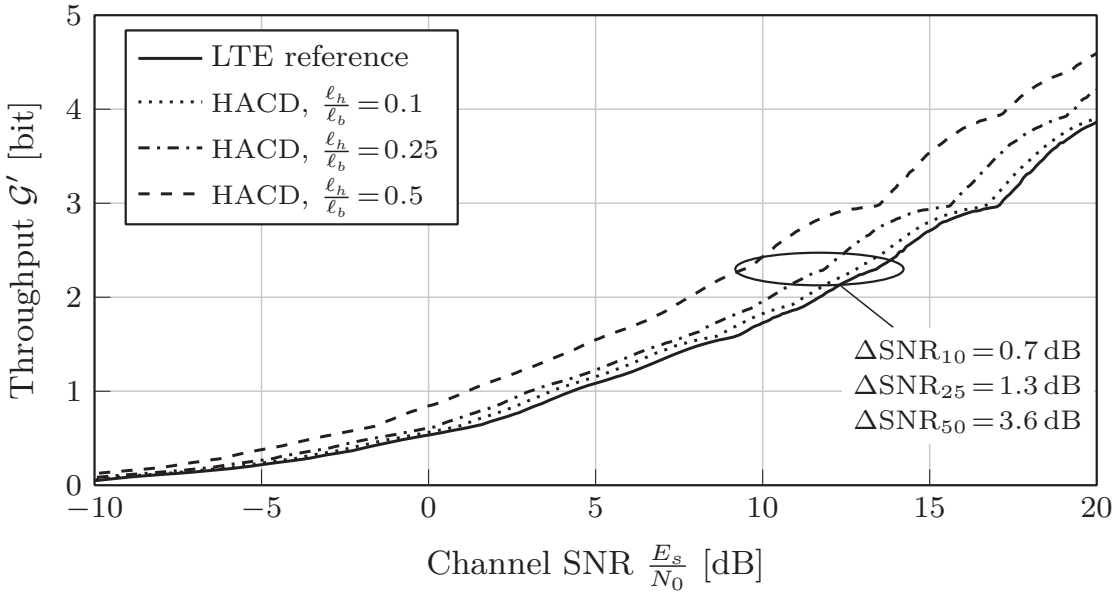


Figure 7.5: Scaled throughput $\mathcal{G}' = \ell_b/\ell_v \cdot \mathcal{G}$ of LTE employing HACD, illustrated for the header fractions $\ell_h/\ell_b \in \{0.1, 0.25, 0.5\}$ and the MCSs given in Table 7.1. The transmission channel is either modeled by an AWGN channel (a) or the 3GPP TU@20 MHz channel (b).

in the network stack which may necessitate an additional header FEC to warranty for robust header recovery. The effect on the spectral efficiency is not obvious in this case. On the one hand, this increases the undesired overhead, but on the other hand, it also boosts the performance of HACD, since a higher fraction of known bits can be exploited. The loss in spectral efficiency can, thus, at least partly be

compensated. However, a detailed evaluation of this interaction is out of the scope of this thesis.

There are other multimedia transmission scenarios, in which RoHC is only optional, although the header may consume a considerable fraction of the packet. The G.719 audio codec, standardized by the *ITU Telecommunication Standardization Sector* (ITU-T), e.g., generates frames between 80 bytes and 320 bytes [WJ09] resulting in a header fraction (without compression) of $11\,\% \leq \ell_h/\ell_b \leq 33\,\%$ (no frame aggregation considered). This illustrates that the evaluated header fractions are subject to absolutely reasonable assumptions.

As mentioned above, the predicted gains are bounds on the performance of HACD assuming perfect header decoding. To approach these bounds, system modification may be required. In HACD, *extrinsic* information can only be fed back if the header have been retrieved without errors. Besides the utilization of the SDHD principle, this requires a strong header FEC. Better system performance can be expected if *Iterative Header-Channel Decoding* (IHCD) is carried out. Similar to ISCD, (imperfect) *extrinsic* information is exchanged between the SDHD decoder and the channel decoder in an iterative process. As a result, the reliability of the header bits and the payload bits are jointly improved. A first successful attempt has been presented in [JMZD09] for physical layer headers, however, this concept can easily be enhanced to the complete network stack. Therefore, the derived performance bounds are still valid.

Chapter 8

Summary

This thesis has addressed state-of-the-art as well as novel rate matching (adaptive coding) schemes for wireless and mobile (cellular) communication. In the LTE standard for mobile communication, the required higher or lower code rates can be obtained by bit puncturing or bit repetition in the output stream of a fixed-rate mother Turbo code. However, a high potential alternative to bit repetition is dummy bit insertion, in which known bits are inserted into the information bit sequence before channel encoding. The emphasis of this work has been on the information theoretic analysis of dummy bit insertion, its performance comparison to bit repetition, and its integration into the LTE HARQ scheme. Furthermore, the potential of dummy bit insertion for systems employing LDPC codes has been demonstrated.

In the first part, information theoretic transmission models have been derived for all considered rate matching schemes to allow for a semi-analytical evaluation of the system behavior by means of EXIT charts. The presented models cover parallel concatenated Turbo codes, serially concatenated Turbo codes, as well as LDPC codes. It has been illustrated in all cases that the analytically derived EXIT charts perfectly match their measured versions if the rate matching schemes are modeled by random processes and if the communication channel can be modeled as BEC. Assuming an AWGN channel or a BSC as communication channel, at least very close approximations have been obtained. The obtained expressions enable a semi-analytical rate matching procedure, in which the desired number of punctured, repeated or inserted bits can be determined analytically for all target channel qualities based on the EXIT chart simulation of the underlying mother Turbo code. Extensive EXIT chart simulations are, thus, superfluous. A performance comparison between bit repetition and dummy bit insertion by means of their analytically determined EXIT charts has further been carried out for the LTE mother Turbo code. In case of AWGN, dummy bit insertion is more effective than bit repetition for code rates between 1/5 and 1/3 (mother code rate). Gains up to 0.5 dB in terms of channel SNR have been achieved.

However, in standardized systems as, e.g., LTE, bit puncturing and repetition is carried out according to a predetermined (deterministic) pattern. Therefore, the general transmission models have been refined accordingly, leading to very close

approximations of the actual (simulated) EXIT charts. This observation holds also true for optimal dummy bit insertion, in which known bits are *equidistantly* (not randomly) inserted into the information bit sequence before encoding.

The theoretical results have then been confirmed by the evaluation of the modified LTE system employing optimal dummy bit insertion and the standardized LTE system using bit repetition. Both systems have been compared in terms of their residual FER, convergence speed, and maximum system throughput for different modulation and coding schemes. The novel system has shown clear performance improvements in all evaluated scenarios. Besides the gains of up to 0.45 dB in terms of channel SNR observed by FER and throughput simulations, a considerable convergence speedup is realized by dummy bit insertion. While the standard LTE systems requires 20–30 Turbo iterations for full convergence, only 10 iterations are sufficient with the novel system.

Moreover, dummy bit insertion for LDPC codes has briefly been addressed in this thesis. The "optimal" insertion of dummy bits into the information bit sequence is not as trivial as for convolutional Turbo codes, in which the dummy bits should be equally spaced within the input frame. The proposed algorithm selects the dummy bit positions within the frame according to their distance within the Tanner graph. The optimization constraint is to maximize the average distance of all dummy bits, resulting in a set of positions with approximately equal distance to each other. Simulation for the WiMAX systems have illustrated that comparable FER performance can be expected for the novel system using the rate-5/6 WiMAX LDPC code as fixed-rate mother code and employing dummy bit insertion for code rate adaptation. However, the modified system benefits from a considerably increased code rate flexibility and hardware efficiency, since only one encoder/decoder pair has to be implemented.

All of the above evaluations have shown the potential of dummy bit insertion for mobile communication systems. However, these systems will employ HARQ schemes, in which the transmission of additional redundancy regarding the same data frame is permitted if residual bit errors are detected at the receiver after channel decoding. The integration of dummy bit insertion into such a system is challenging, since the number of inserted dummy bits has to be fixed before channel encoding and cannot be controlled during HARQ. However, a fundamental property of dummy bit insertion, the *puncturing property*, has been proven in this thesis which enables the control of the effective code rate after channel encoding by means of dummy bit insertion. This property states that each inserted dummy bit is fully transparent for the channel decoder if all corresponding parity bits are punctured. Accordingly, in the proposed novel LTE HARQ scheme, 50 % dummy bits are initially inserted before rate-1/3 Turbo encoding. While higher target code rates are still generated by means of bit puncturing, code rates between 1/5 and 1/3 are provided by dummy bit insertion, exploiting the above described puncturing property. Only code rates lower than 1/5 are generated by bit repetition. Based

on the standard LTE system, a modified ring buffer implementation is used for rate matching. Evaluations of the system throughput have revealed that the novel LTE HARQ system outperforms the standardized system in a wide range of channel conditions. Gains up to 1.4 dB have been realized depending on the selected transmission channel model. In particular if additional HARQ transmissions are carried out, the benefit of the novel HARQ system is clearly visible. Only minor transmitter and receiver modifications are necessary to achieve these performance improvements.

Furthermore, HARQ with unreliable ACK/NACK feedback has been considered in this thesis. The influence of signaling errors on the overall system throughput has been analytically derived for arbitrary communication systems. The presented results will support the specification of tolerable ACK/NACK error probabilities and, thus, the design of the error protection scheme used on the feedback link. Simulations of the standard LTE HARQ system with reliable and unreliable feedback have validated the analytical results.

In the last part of this thesis, the EXIT chart expressions derived for dummy bit insertion have been applied to approximate upper bounds on the performance of *Header bit Assisted Channel Decoding* (HACD), in which perfect *a priori* information about error-free decoded headers is fed back to the channel decoder. For header fractions of 10 %, 25 %, or 50 %, gains up to 0.36 dB, 0.95 dB, or 2.25 dB can be achieved. Note that header fractions of 20 % to 40 % can be found in state-of-the-art systems even if header compression is used. However, sophisticated HACD system designs have to be developed in order to approach the theoretical bounds.

In summary, it can be stated that rate matching by dummy bit insertion and the concept of HACD are both high potential candidates for future improvements of LTE communication systems. They account for the rapidly increasing data rates by significantly improving the spectral efficiency of the system at the price of a moderately increased complexity and energy consumption.

Appendix A

Channel Models

Four specific channel models are considered in this thesis, namely the *Binary Erasure Channel* (BEC), the *Binary Symmetric Channel* (BSC), the (fading) *Additive White Gaussian Noise Channel* (AWGNC) and the *Typical Urban* (TU) channel model specified by the *3rd Generation Partnership Project* (3GPP).

A.1 Binary Erasure Channel

The channel model of the BEC is depicted in Figure A.1. It defines a mapping $\mathfrak{U} \mapsto \mathfrak{V}$: $\mathbb{F}_2 \to \mathbb{V} = \{0, 1, *\}$ in which the output symbol $*$ denotes an erasure. For distinct realizations $\mathfrak{U} = \mathfrak{u}$ and $\mathfrak{V} = \mathfrak{v}$, the transition probabilities are given according to

$$p_{\mathfrak{V}|\mathfrak{U}}(\mathfrak{v}|\mathfrak{u}) = \begin{cases} 1 - P_\mathrm{E} & : \quad \mathfrak{v} = \mathfrak{u} \\ P_\mathrm{E} & : \quad \mathfrak{v} = * \\ 0 & : \quad \text{otherwise} \end{cases} \quad , \quad \mathfrak{u} \in \mathbb{F}_2, \mathfrak{v} \in \mathbb{V} \tag{A.1}$$

A fundamental measure of a communication channel is the channel capacity C which is defined as

$$C = \max_{P_\mathfrak{U}(\mathfrak{u})} \mathcal{I}(\mathfrak{U}; \mathfrak{V}) = \max_{P_\mathfrak{U}(\mathfrak{u})} \left(H(\mathfrak{U}) - H(\mathfrak{U}|\mathfrak{V}) \right) = \max_{P_\mathfrak{U}(\mathfrak{u})} \left(H(\mathfrak{V}) - H(\mathfrak{V}|\mathfrak{U}) \right) \tag{A.2}$$

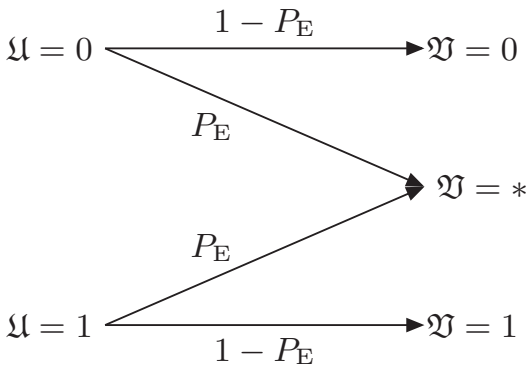


Figure A.1: Binary erasure channel model.

with the entropy

$$H(\mathfrak{V}) = -\sum_{\mathfrak{v}\in\mathbb{V}} P_{\mathfrak{V}}(\mathfrak{V}=\mathfrak{v}) \cdot \log_2(P_{\mathfrak{V}}(\mathfrak{V}=\mathfrak{v})) \tag{A.3}$$

and the conditional entropy

$$H(\mathfrak{V}|\mathfrak{U}) = -\sum_{\mathfrak{u}\in\mathbb{F}_2}\sum_{\mathfrak{v}\in\mathbb{V}} P_{\mathfrak{V}|\mathfrak{U}}(\mathfrak{v}|\mathfrak{u}) \cdot P_{\mathfrak{V}}(\mathfrak{v}) \cdot \log_2\left(P_{\mathfrak{V}|\mathfrak{U}}(\mathfrak{v}|\mathfrak{u})\right). \tag{A.4}$$

The capacity of the BEC is reached for equiprobable input bits $\mathfrak{U}$ resulting in an output distribution

$$p_{\mathfrak{V}}(\mathfrak{v}) = \begin{cases} \frac{1-P_{\mathrm{E}}}{2} & : \quad \mathfrak{v}\in\mathbb{F}_2 \\ P_{\mathrm{E}} & : \quad \mathfrak{v}=* \end{cases}. \tag{A.5}$$

The capacity of the BEC is then given by means of (A.1) - (A.5) according to

$$\begin{aligned} C_{\mathrm{BEC}} &= -2\cdot\frac{1-P_{\mathrm{E}}}{2}\cdot\log_2\left(\frac{1-P_{\mathrm{E}}}{2}\right) - P_{\mathrm{E}}\cdot\log_2 P_{\mathrm{E}} \\ &\quad + 2\cdot\frac{1}{2}\cdot(1-P_{\mathrm{E}})\cdot\log_2(1-P_{\mathrm{E}}) + 2\cdot\frac{1}{2}\cdot P_{\mathrm{E}}\cdot\log_2 P_{\mathrm{E}} \\ &= 1-P_{\mathrm{E}}. \end{aligned} \tag{A.6}$$

If the channel is capable to output reliability information, so-called channel-related *a posteriori Log-Likelihood Ratio*s (LLRs) can be defined according to [HOP96]:

$$L(\mathfrak{u}|\mathfrak{v}) = \ln\left(\frac{p_{\mathfrak{U}|\mathfrak{V}}(\mathfrak{u}=0|\mathfrak{V}=\mathfrak{v})}{p_{\mathfrak{U}|\mathfrak{V}}(\mathfrak{u}=1|\mathfrak{V}=\mathfrak{v})}\right) \tag{A.7}$$

$$\begin{aligned} &= \ln\left(\frac{p_{\mathfrak{V}|\mathfrak{U}}(\mathfrak{v}|\mathfrak{u}=0)\cdot P_{\mathfrak{U}}(\mathfrak{u}=0)}{p_{\mathfrak{V}|\mathfrak{U}}(\mathfrak{v}|\mathfrak{u}=1)\cdot P_{\mathfrak{U}}(\mathfrak{u}=1)}\right) \\ &= L(\mathfrak{v}|\mathfrak{u}) + L(\mathfrak{u}). \end{aligned} \tag{A.8}$$

For equiprobable bits $\mathfrak{u}$ and using a bipolar representation of $\mathfrak{v}$, i.e. $\mathfrak{v}\in\{-1,+1\}$, (A.8) can be simplified to

$$L(\mathfrak{u}|\mathfrak{v}) = L_{\mathrm{c}}\cdot\mathfrak{v} = \begin{cases} \infty\cdot\mathfrak{v} & : \quad \mathfrak{v}\in\mathbb{F}_2 \\ 0 & : \quad \mathfrak{v}=* \end{cases}. \tag{A.9}$$

The BEC delivers perfect information with probability $1-P_{\mathrm{E}}$ and no information with probability P_{E}.

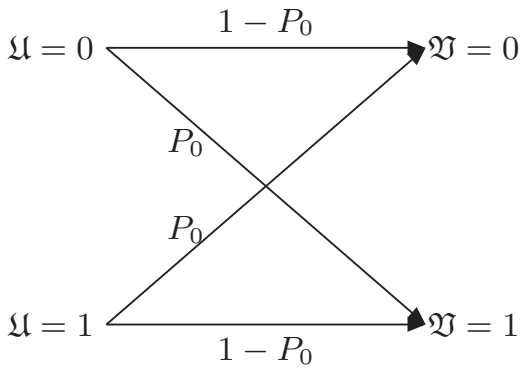


Figure A.2: Binary symmetric channel model.

A.2 Binary Symmetric Channel

The channel model of the BSC, which defines a mapping $\mathfrak{U} \mapsto \mathfrak{V}$: $\mathbb{F}_2 \to \mathbb{F}_2$, is depicted in Figure A.2. For distinct realizations $\mathfrak{U} = \mathfrak{u}$ and $\mathfrak{V} = \mathfrak{v}$, the transition probabilities are dependent on the crossover probability P_0:

$$p_{\mathfrak{V}|\mathfrak{U}}(\mathfrak{v}|\mathfrak{u}) = \begin{cases} 1 - P_0 & : \quad \mathfrak{v} = \mathfrak{u} \\ P_0 & : \quad \mathfrak{v} \neq \mathfrak{u} \end{cases}, \quad \mathfrak{u} \in \mathbb{F}_2, \mathfrak{v} \in \mathbb{F}_2. \tag{A.10}$$

The capacity of a BSC with equiprobable input bits is then given by means of (A.2) according to

$$\begin{aligned} C_{\mathrm{BSC}} &= 1 - H(\mathfrak{V}|\mathfrak{U}) = 1 - H'(P_0) \\ &= 1 + P_0 \cdot \log_2 P_0 + (1 - P_0) \cdot \log_2(1 - P_0) \end{aligned} \tag{A.11}$$

with

$$H'(P_0) = -P_0 \cdot \log_2 P_0 - (1 - P_0) \cdot \log_2(1 - P_0) \tag{A.12}$$

commonly denoted as *binary entropy function* for uniformly distributed (equiprobable) input bits.

By means of (A.8), *a posteriori* LLRs can also be calculated for the BSC. They are only dependent on the crossover probability P_0:

$$L(\mathfrak{u}|\mathfrak{v}) = L_{\mathrm{c}} \cdot \mathfrak{v} \quad \text{with} \quad L_{\mathrm{c}} = \ln\left(\frac{1 - P_0}{P_0}\right). \tag{A.13}$$

A.3 Additive White Gaussian Noise Channel

Let $\mathfrak{U}$, $\mathfrak{V}$, $\mathfrak{N}$ be random variables with $\mathfrak{V} = \mathfrak{U} + \mathfrak{N}$ and $\mathfrak{N} \sim \mathcal{N}(0, \sigma_n^2)$, i.e., $\mathfrak{N}$ is Gaussian distributed with zero mean and variance σ_n^2 according to

$$p_{\mathfrak{N}}(\mathfrak{n}) = \frac{1}{\sqrt{2\,\pi\,\sigma_n^2}} \cdot \exp\left(-\frac{\mathfrak{v}^2}{2\,\sigma_n^2}\right). \tag{A.14}$$

Then, the capacity of the AWGN channel is given by

$$C_{\mathrm{AWGNC}} = \max_{P_{\mathfrak{U}}(\mathfrak{u})} \mathcal{I}(\mathfrak{U};\mathfrak{V}) = \frac{1}{2}\log_2\left(1+\frac{\sigma_u^2}{\sigma_n^2}\right), \tag{A.15}$$

since $\mathcal{I}(\mathfrak{U};\mathfrak{V})$ is maximized for $\mathfrak{U} \sim \mathcal{N}(0,\sigma_u^2)$ [CT91].

For BPSK transmission of equiprobable, bipolar input bits $\mathfrak{u} \in \{-1,1\}$, the input distribution is given by $P_{\mathfrak{U}}(\mathfrak{u}=-1) = P_{\mathfrak{U}}(\mathfrak{u}=1) = 1/2$. Then, the mutual information $\mathcal{I}(\mathfrak{U};\mathfrak{V})$ can be computed to [Moo05]

$$C_{\mathrm{BIAWGNC}} = \mathcal{I}(\mathfrak{U};\mathfrak{V}) = -\int_{-\infty}^{\infty} P_{\mathfrak{V}}(\mathfrak{v})\cdot\log_2 P_{\mathfrak{V}}(\mathfrak{v})\,\mathrm{d}\mathfrak{v} - \frac{1}{2}\log_2\left(2\,\pi\,\mathrm{e}\,\sigma_n^2\right), \tag{A.16}$$

with

$$P_{\mathfrak{V}}(\mathfrak{v}) = \frac{1}{\sqrt{8\,\pi\,\sigma_n^2}}\cdot\sum_{\mathfrak{u}=-1,+1}\exp\left(-\frac{(\mathfrak{v}-\mathfrak{u})^2}{2\,\sigma_n^2}\right). \tag{A.17}$$

The first term in (A.16) has to be numerically evaluated. It can further be shown that $C_{\mathrm{AWGNC}} > C_{\mathrm{BIAWGNC}}$.

From Shannon's coding theorem, it is known that reliable transmission (without errors) is only possible if the transmission rate R is lower than the channel capacity C. However, if a residual bit error rate unequal to zero is allowed by the system, the maximum achievable transmission rate can exceed the channel capacity. Assuming that statistically independent bit errors occur with probability P_0, the effective transmission rate R_{P} is reduced by the factor of $C_{\mathrm{BSC}} = 1 - H'(P_0)$ as it is known from rate-distortion theory [Ber71]. By means of Shannon's coding theorem it can finally be concluded that

$$R_{\mathrm{P}} = \frac{C}{1-H'(P_0)}. \tag{A.18}$$

For the example of channel coding with code rate r_{C} and BPSK modulation, (A.18) can be transformed to

$$r_{\mathrm{C}} = \frac{C_{\mathrm{BIAWGNC}}}{1-H'(P_0)}, \tag{A.19}$$

which provides the desired relation between the bit error rate and the channel quality. This defines the Shannon bound for BPSK transmission with errors.

By means of (A.8), (A.14), and $E_s/N_0 = 1/(2\,\sigma_n^2)$, LLRs can also be calculated for the fading BIAWGNC with fading factor a:

$$\begin{aligned} L(\mathfrak{u}|\mathfrak{v}) = L(\mathfrak{v}|\mathfrak{u}) &= \ln\left(\frac{p_{\mathfrak{V}|\mathfrak{U}}(\mathfrak{v}|\mathfrak{u}=0)}{p_{\mathfrak{V}|\mathfrak{U}}(\mathfrak{v}|\mathfrak{u}=1)}\right) = \ln\left(\frac{p_{\mathfrak{N}}(\mathfrak{v}-a)}{p_{\mathfrak{N}}(\mathfrak{v}+a)}\right) \\ &= \ln\left(\frac{\exp\left(-\frac{E_s}{N_0}(\mathfrak{v}-a)^2\right)}{\exp\left(-\frac{E_s}{N_0}(\mathfrak{v}+a)^2\right)}\right) = 4\,a\frac{E_s}{N_0}\cdot\mathfrak{v} = L_{\mathrm{c}}\cdot\mathfrak{v}. \end{aligned} \tag{A.20}$$

A.4 3GPP Typical Urban Channel Model

The *Typical Urban* (TU) channel model used in this work is based on the model specified by 3GPP in [3GP08] and adapted to the LTE physical layer according to [Lüd10]. It consists of a tapped delay line model defined by their delay and average inverse attenuation. In [Lüd10], the adaptation is carried out for the system bandwidth 5 MHz and 20 MHz with symbol duration $T_s^{-1} = 512 \cdot 15\,\text{kHz}$ and $T_s^{-1} = 2048 \cdot 15\,\text{kHz}$, respectively. The symbol duration is determined by the number of subcarriers (512 or 2048) and their bandwidth (15 kHz). In this work, only the LTE system with 20 MHz bandwidth is considered. For this case, the TU model derived in [Lüd10] is given by an FIR filter with 66 delay taps:

Table A.1: Typical urban channel model for an LTE system with 20 MHz system bandwidth.

FIR tap delay in T_s	Attenuation in dB
0	1.08
7	3.07
16	5.58
21	6.69
27	8.76
38	11.76
40	12.41
41	12.67
47	14.38
50	15.23
56	16.91
58	17.52
60	18.07
63	18.91
66	19.75

Appendix B

Mathematical Derivations

Proof of Lemma 4.2.1

According to [AKtB04], the *extrinsic* information $\mathcal{I}_{\mathrm{C}}^{\Sigma}$ generated by deterministic bit repetition with code rate $r_{\mathrm{DR}} = 1/Q$ and for a transmission over a BEC with mutual information $\mathcal{I}_{\mathrm{C}}$ is given by

$$\mathcal{I}_{\mathrm{C}}^{\Sigma} = 1 - (1 - \mathcal{I}_{\mathrm{C}})^{Q} = 1 - \prod_{q=1}^{Q} (1 - \mathcal{I}_{\mathrm{C}}). \tag{B.1}$$

The puncturing effect can now be incorporated by substituting $\mathcal{I}_{\mathrm{C}}$ in each branch with the *equivalent* mutual information $\mathcal{I}_{\mathrm{C}}^{(q)} = r_{\mathrm{RP}}^{(q)} \cdot \mathcal{I}_{\mathrm{C}}$:

$$\mathcal{I}_{\mathrm{C}}^{\Sigma} = 1 - \prod_{q=1}^{Q} \left(1 - \mathcal{I}_{\mathrm{C}}^{(q)}\right). \tag{B.2}$$

This proves Lemma 4.2.1.

Proof of Lemma 4.2.2

Instead of maximizing (4.21), we minimize the function

$$\Phi(\mathbf{r}) = 1 - \mathcal{I}_{\mathrm{C}}^{\Sigma} = \prod_{q=1}^{Q} \left(1 - r_{\mathrm{RP}}^{(q)} \cdot \mathcal{I}_{\mathrm{C}}\right) \quad , \quad \mathbf{r} = \left(r_{\mathrm{RP}}^{(1)} \ldots r_{\mathrm{RP}}^{(Q)}\right)^{T} \tag{B.3}$$

subject to

$$r_{\mathrm{RP}} = \frac{1}{Q} \sum_{q=1}^{Q} r_{\mathrm{RP}}^{(q)}, \quad r_{\mathrm{RP}}^{(1)} = 1, \quad 0 \leq r_{\mathrm{RP}}^{(q)} \leq 1, \tag{B.4}$$

where $0 \leq r_{\mathrm{RP}} \leq 1$ denotes the average puncturing rate which is arbitrary within the domain but fixed. Substitution of $r_{\mathrm{RP}}^{(Q)}$ in (B.3) by means of (B.4) results in

$$\Phi\left(\mathbf{r}^{\setminus Q}\right) = \left(1 - \left(Q \cdot r_{\mathrm{RP}} - \sum_{q=1}^{Q-1} r_{\mathrm{RP}}^{(q)}\right) \cdot \mathcal{I}_{\mathrm{C}}\right) \cdot \prod_{q=1}^{Q-1} \left(1 - r_{\mathrm{RP}}^{(q)} \cdot \mathcal{I}_{\mathrm{C}}\right) \tag{B.5}$$

with $\mathbf{r}^{\setminus Q} = \left(r_{\mathrm{RP}}^{(1)} \dots r_{\mathrm{RP}}^{(Q-1)}\right)$. In order to determine all local extrema for functions of more than one variable, all first partial derivatives have to be zero. Due to the symmetry of $\Phi\left(\mathbf{r}^{\setminus Q}\right)$ with respect to each $r_{\mathrm{RP}}^{(q)}$, the first derivative of all variables is given according to

$$\begin{aligned}\frac{\partial \Phi\left(\mathbf{r}^{\setminus Q}\right)}{\partial r_{\mathrm{RP}}^{(\eta)}} = &\left[\prod_{\substack{q=1\\q\neq\eta}}^{Q-1}\left(1 - r_{\mathrm{RP}}^{(q)} \cdot \mathcal{I}_{\mathrm{C}}\right)\right] \cdot \left[\mathcal{I}_{\mathrm{C}} \cdot \left(1 - r_{\mathrm{RP}}^{(\eta)} \cdot \mathcal{I}_{\mathrm{C}}\right)\right.\\ &\left. - \mathcal{I}_{\mathrm{C}} \cdot \left(1 - \mathcal{I}_{\mathrm{C}} \cdot \left(Q \cdot r_{\mathrm{RP}} - \sum_{q=1}^{Q-1} r_{\mathrm{RP}}^{(q)}\right)\right)\right] \overset{!}{=} 0 \end{aligned} \tag{B.6}$$

$$\Leftrightarrow 1 - r_{\mathrm{RP}}^{(\eta)} \cdot \mathcal{I}_{\mathrm{C}} - 1 + Q \cdot r_{\mathrm{RP}} \cdot \mathcal{I}_{\mathrm{C}} - \mathcal{I}_{\mathrm{C}} \sum_{q=1}^{Q-1} r_{\mathrm{RP}}^{(q)} = 0$$

$$\Leftrightarrow 1 - r_{\mathrm{RP}}^{(\eta)} \cdot \mathcal{I}_{\mathrm{C}} - 1 + Q \cdot r_{\mathrm{RP}} \cdot \mathcal{I}_{\mathrm{C}} - \mathcal{I}_{\mathrm{C}} - \mathcal{I}_{\mathrm{C}} \sum_{q=2}^{Q-1} r_{\mathrm{RP}}^{(q)} = 0$$

$$\Leftrightarrow 2\, r_{\mathrm{RP}}(\eta) + \sum_{\substack{q=2\\q\neq\eta}}^{Q-1} r_{\mathrm{RP}}^{(q)} = Q \cdot r_{\mathrm{RP}} - 1 \quad , \quad 2 \leq \eta \leq Q-1 \quad , \quad \mathcal{I}_{\mathrm{C}} \neq 0. \tag{B.7}$$

Equation (B.7) in conjunction with (B.4) defines a system of linear equations which can be formulated in the following vector notation:

$$\begin{pmatrix} 2 & 1 & \cdots & 1 & 0 \\ 1 & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & 1 & \vdots \\ 1 & \cdots & 1 & 2 & 0 \\ 1 & \cdots & \cdots & \cdots & 1 \end{pmatrix} \cdot \mathbf{r}^{\setminus 1} = (Q \cdot r_{\mathrm{RP}} - 1) \cdot \mathbf{1} \tag{B.8}$$

$$\Leftrightarrow \mathbf{G}_{Q-1} \cdot \mathbf{r}^{\setminus 1} = (Q \cdot r_{\mathrm{RP}} - 1) \cdot \mathbf{1} \tag{B.9}$$

with $\mathbf{r}^{\setminus 1} = \left(r_{\mathrm{RP}}^{(2)} \dots r_{\mathrm{RP}}^{(Q)}\right)$ and $\mathbf{1} = (1 \dots 1)^T$ being a vector of length $(Q-1)$. Since the Matrix $\mathbf{G}_{Q-1}$ is of dimension $(Q-1) \times (Q-1)$, this system consists of $(Q-1)$ equations and $(Q-1)$ variables, exhibiting at most one solution under the given constraints. It is easy to prove that the system is solved for

$$r_{\mathrm{RP}}^{(q)} = \frac{Q \cdot r_{\mathrm{RP}} - 1}{Q-1} \quad , \quad 2 \leq q \leq Q. \tag{B.10}$$

However, this is only a necessary condition for the existence of an extremum. The sufficient condition for a maximum and a minimum can be verified by means of the Hesse matrix which contains all permutations of the second partial derivatives. If the Hesse matrix is negative definite there exist a local maximum at the considered position, if it is positive definite this position corresponds to a local minimum. The second partial derivatives are given by

$$\begin{aligned}\frac{\partial^2\Phi(\mathbf{r}^{\setminus Q})}{\partial r_{\mathrm{RP}}^{(\eta)}\partial r_{\mathrm{RP}}^{(\psi)}} &= -\mathcal{I}_{\mathrm{C}}^2\cdot\prod_{\substack{q=1\\q\neq\psi}}^{Q-1}\left(1-r_{\mathrm{RP}}^{(q)}\cdot\mathcal{I}_{\mathrm{C}}\right)\\ &\quad+\mathcal{I}_{\mathrm{C}}^2\cdot\left(1-\mathcal{I}_{\mathrm{C}}\cdot\left(Q\cdot r_{\mathrm{RP}}-\sum_{q=1}^{Q-1}r_{\mathrm{RP}}^{(q)}\right)\right)\cdot\prod_{\substack{q=1\\q\neq\psi\\q\neq\eta}}^{Q-1}\left(1-r_{\mathrm{RP}}^{(q)}\cdot\mathcal{I}_{\mathrm{C}}\right)\\ &\quad-\mathcal{I}_{\mathrm{C}}^2\cdot\prod_{\substack{q=1\\q\neq\eta}}^{Q-1}\left(1-r_{\mathrm{RP}}^{(q)}\cdot\mathcal{I}_{\mathrm{C}}\right),\quad \eta\neq\psi, \end{aligned}\tag{B.11}$$

$$\frac{\partial^2\Phi(\mathbf{r}^{\setminus Q})}{\partial\left(r_{\mathrm{RP}}^{(\eta)}\right)^2} = -2\,\mathcal{I}_{\mathrm{C}}^2\cdot\prod_{\substack{q=1\\q\neq\eta}}^{Q-1}\left(1-r_{\mathrm{RP}}^{(q)}\cdot\mathcal{I}_{\mathrm{C}}\right).\tag{B.12}$$

Substituting $r_{\mathrm{RP}}^{(q)}$ according to (B.10) results in

$$\frac{\partial^2\Phi(\mathbf{r}^{\setminus Q})}{\partial r_{\mathrm{RP}}^{(\eta)}\partial r_{\mathrm{RP}}^{(\psi)}} = -\mathcal{I}_{\mathrm{C}}^2(1-\mathcal{I}_{\mathrm{C}})\left(1-\frac{Qr_{\mathrm{RP}}-1}{Q-1}\mathcal{I}_{\mathrm{C}}\right)^{Q-3}\leq 0,\quad \eta\neq\psi,\tag{B.13}$$

$$\frac{\partial^2\Phi(\mathbf{r}^{\setminus Q})}{\partial\left(r_{\mathrm{RP}}^{(\eta)}\right)^2} = -2\,\mathcal{I}_{\mathrm{C}}^2(1-\mathcal{I}_{\mathrm{C}})\left(1-\frac{Qr_{\mathrm{RP}}-1}{Q-1}\mathcal{I}_{\mathrm{C}}\right)^{Q-3}\leq 0\tag{B.14}$$

under the constraints of $0<r_{\mathrm{RP}}^{(q)}\leq 1$ $(1\leq q\leq Q)$ and $0<\mathcal{I}_{\mathrm{C}}\leq 1$. Equality in (B.13) and (B.14) is only reached for $\mathcal{I}_{\mathrm{C}}=1$. Consequently, the Hesse matrix is negative definite for all other (relevant) cases. The root of the first derivatives is, thus, associated with a local maximum. However, the objective is to minimize (B.3) in order to obtain the optimal random puncturing. This implicates that the desired minimum is located at the boundary of the considered domains $0<r_{\mathrm{RP}}^{(q)}\leq 1$.

Only a subset of all boundaries of the polytope has to be considered due to the symmetry of $\Phi(\mathbf{r})$ in $r_{\mathrm{RP}}^{(q)}$ $(1\leq q\leq Q)$. This set is given according to

$$\mathbf{r}_{\zeta}^{\setminus 1}=\left(\underbrace{1\ldots 1}_{(\zeta-1)\text{-times}},r_{\mathrm{RP}}^{(\zeta+1)},\ldots,r_{\mathrm{RP}}^{(Q)}\right),\quad 2\leq\zeta\leq\lceil Q\cdot r_{\mathrm{RP}}\rceil-1=Q'-1.\tag{B.15}$$

The restriction to $\zeta\leq Q'-1$ is due to Equation (B.4). For an analysis of the boundaries, the order of the system of linear equations, which is given in (B.9),

can be reduced by ζ with $2 \leq \zeta \leq Q' - 1$:

$$\mathbf{G}_{Q-\zeta} \cdot \mathbf{r}^{\setminus\{1\dots\zeta\}} = (Q \cdot r_{\mathrm{RP}} - \zeta) \cdot \mathbf{1} \tag{B.16}$$

with $\mathbf{r}^{\setminus\{1\dots\zeta\}} = \left(r_{\mathrm{RP}}^{(\zeta+1)} \dots r_{\mathrm{RP}}^{(Q)}\right)$. The system of equations is solved in dependence on ζ by

$$r_{\mathrm{RP}}^{(q)} = \frac{Q \cdot r_{\mathrm{RP}} - \zeta}{Q-1} \quad , \quad \zeta + 1 \leq q \leq Q. \tag{B.17}$$

Note that (B.10) is included for $\zeta = 1$. This solution corresponds to a local maximum, i.e., $\Phi(\mathbf{r})$ is concave at all boundaries of the considered polytope.

Therefore, it can be concluded that the global minimum is located at the last boundary

$$\mathbf{r}_{Q'-1}^{\setminus 1} = \left(\underbrace{1 \dots 1}_{(Q'-1)\text{-times}}, r_{\mathrm{RP}}^{(Q')}, \dots, r_{\mathrm{RP}}^{(Q)} \right) \tag{B.18}$$

and reached for

$$r_{\mathrm{RP}}^{(q)} = \begin{cases} 1 & , \quad 1 \leq q < Q' \\ Q \cdot r_{\mathrm{RP}} - (Q'-1) & , \quad q = Q' \\ 0 & , \quad Q' < q \leq Q \end{cases} \quad , \quad Q' = \lceil Q \cdot r_{\mathrm{RP}} \rceil \tag{B.19}$$

due to (B.4).

Proof of Proposition 4.3.1

According to the chain rule for mutual information [CT91], the mutual information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],\star}$ at the output of two parallel concatenated channels with mutual information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],1} = \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]}$ and $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],2} = P_{\mathrm{D}}$, respectively, is given by

$$\begin{aligned} \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],\star} &:= \mathcal{I}\left(\mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],\star}(x); X\right) = \mathcal{I}\left(\mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],1}(x), \mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],2}(x); X\right) \\ &= \mathcal{I}\left(\mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],1}(x); X\right) + \mathcal{I}\left(\mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],2}(x); X\right) \\ &\quad - \mathcal{I}\left(\mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],1}(x); \mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],2}(x)\right) \qquad \text{(B.20)} \\ &= \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],1} + \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],2} - \mathcal{I}\left(\mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],1}(x); \mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],2}(x)\right) \qquad \text{(B.21)} \\ &= \mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}]} + P_{\mathrm{D}} - \mathcal{I}\left(\mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],1}(x); \mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],2}(x)\right) \qquad \text{(B.22)} \end{aligned}$$

The term $\mathcal{I}\left(\mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],1}(x); \mathcal{L}_{\mathrm{CD}}^{[\mathrm{apri}],2}(x)\right)$ is called information defect and can be calculated for a parallel concatenation of a BEC with mutual information $\mathcal{I}_{\mathrm{CD}}^{[\mathrm{apri}],2} =$

P_D and an arbitrary channel with mutual information $\mathcal{I}_\mathrm{CD}^{[\mathrm{apri}],1} = \mathcal{I}_\mathrm{CD}^{[\mathrm{apri}]}$. The information defect comprises the information which is delivered jointly by both channels. In the case of the BEC, perfect information is provided for a fraction of P_D bits. Hence, it can be concluded that all information which is delivered by the other channel about that fraction of bits has to be subtracted. This leads directly to the desired expression of the information defect:

$$\mathcal{I}\left(\mathcal{L}_\mathrm{CD}^{[\mathrm{apri}],1}(x); \mathcal{L}_\mathrm{CD}^{[\mathrm{apri}],2}(x)\right) = \mathcal{I}_\mathrm{CD}^{[\mathrm{apri}],1} \cdot \mathcal{I}_\mathrm{CD}^{[\mathrm{apri}],2} = \mathcal{I}_\mathrm{CD}^{[\mathrm{apri}]} \cdot P_\mathrm{D}. \tag{B.23}$$

The above considerations are valid due to the random nature of the dummy bit insertion and proof Proposition 4.3.1:

$$\begin{aligned}\mathcal{I}_\mathrm{CD}^{[\mathrm{apri}],\star} &= \mathcal{I}_\mathrm{CD}^{[\mathrm{apri}]} + P_\mathrm{D} - P_\mathrm{D} \cdot \mathcal{I}_\mathrm{CD}^{[\mathrm{apri}]} \\ &= \mathcal{I}_\mathrm{CD}^{[\mathrm{apri}]} + P_\mathrm{D} \cdot (1 - \mathcal{I}_\mathrm{CD}^{[\mathrm{apri}]}) \geq \mathcal{I}_\mathrm{CD}^{[\mathrm{apri}]}. \end{aligned} \tag{B.24}$$

Proof of Proposition 4.5.1

In general, the mutual information at the output of the effective channel depicted in Figure 4.22 is given according to [CT91]

$$\mathcal{I}^\mathrm{ser}(\mathfrak{W}; \mathfrak{U}) = H(\mathfrak{W}) - H(\mathfrak{W}|\mathfrak{U}), \tag{B.25}$$

where $H(\cdot)$ and $H(\cdot|\cdot)$ denote the entropy and the conditional entropy of the output bits, respectively.

The first term can be computed for a realization $\mathfrak{U} = \mathfrak{u}$ and $\mathfrak{W} = \mathfrak{w}$ and for a uniformly distributed input source $P_\mathfrak{U}(\mathfrak{u}) = 1/2$ by means of the transition probabilities $P_{\mathfrak{W}|\mathfrak{U}}(\mathfrak{w}|\mathfrak{u}) \in \{P_\mathrm{E}, (1-P_\mathrm{E}) \cdot P_0, (1-P_\mathrm{E}) \cdot (1-P_0)\}$:

$$H(\mathfrak{W}) = -\sum_{\mathfrak{w} \in \mathbb{W}} P_\mathfrak{W}(\mathfrak{w}) \cdot \log_2(P_\mathfrak{W}(\mathfrak{w})) \tag{B.26}$$

$$P_\mathfrak{W}(\mathfrak{w}) = -\sum_{\mathfrak{u} \in \mathbb{F}_2} P_\mathfrak{U}(\mathfrak{u}) \cdot P_{\mathfrak{W}|\mathfrak{U}}(\mathfrak{w}|\mathfrak{u}) \tag{B.27}$$

resulting in

$$H(\mathfrak{W}) = -(1-P_\mathrm{E}) \cdot \log_2\left(\frac{1-P_\mathrm{E}}{2}\right) - P_\mathrm{E} \cdot \log_2(P_\mathrm{E}). \tag{B.28}$$

The latter term in (B.25) is given by

$$H(\mathfrak{W}|\mathfrak{U}) = \sum_{\mathfrak{u} \in \mathbb{F}_2} \sum_{\mathfrak{w} \in \mathbb{W}} P_\mathfrak{U}(\mathfrak{u}) \cdot P_{\mathfrak{W}|\mathfrak{U}}(\mathfrak{w}|\mathfrak{u}) \cdot \log_2(P_{\mathfrak{W}|\mathfrak{U}}(\mathfrak{w}|\mathfrak{u})) \tag{B.29}$$

$$\begin{aligned} &= -P_\mathrm{E} \cdot \log_2(P_\mathrm{E}) - (1-P_\mathrm{E}) \cdot P_0 \cdot \log_2((1-P_\mathrm{E}) \cdot P_0) \\ &\quad - (1-P_\mathrm{E}) \cdot (1-P_0) \cdot \log_2((1-P_\mathrm{E}) \cdot (1-P_0)) \end{aligned} \tag{B.30}$$

Substituting (B.28) and (B.30) in (B.25) proofs Proposition 4.5.1:

$$\begin{aligned}
\mathcal{I}^{\mathrm{ser}}(\mathfrak{W};\mathfrak{U}) &= -(1-P_{\mathrm{E}})\cdot\log_2\left(\frac{1-P_{\mathrm{E}}}{2}\right) \\
&\quad + (1-P_{\mathrm{E}})\cdot P_0\cdot\log_2((1-P_{\mathrm{E}})\cdot P_0) \\
&\quad + (1-P_{\mathrm{E}})\cdot(1-P_0)\cdot\log_2((1-P_{\mathrm{E}})\cdot(1-P_0)) \\
&= (1-P_{\mathrm{E}})\cdot[1+P_0\cdot\log_2(P_0)+(1-P_0)\cdot\log_2(1-P_0)] \qquad \text{(B.31)} \\
&= (1-P_{\mathrm{E}})\cdot(1-H'(P_0)) \qquad \text{(B.32)} \\
&= \mathcal{I}_{\mathrm{BEC}}\cdot\mathcal{I}_{\mathrm{BSC}} \qquad \text{(B.33)}
\end{aligned}$$

Proof of Proposition 4.5.2

For any mapping $\mathfrak{U} \mapsto \mathfrak{W} = (\mathfrak{W}_1,\ldots,\mathfrak{W}_q,\ldots,\mathfrak{W}_Q) : \mathbb{F}_2 \to \mathrm{W}^Q$, the mutual information $\mathcal{I}^{\mathrm{par},1}$ is given according to

$$\mathcal{I}^{\mathrm{par},1}(\mathfrak{W};\mathfrak{U}) = H(\mathfrak{W}) - H(\mathfrak{W}|\mathfrak{U}). \qquad \text{(B.34)}$$

The entropy $H(\mathfrak{W})$ can be computed using (a) Bayes and assuming (b) the conditional independence of the channel outputs $\mathfrak{w}_q$ as well as (c) uniformly distributed input bits $\mathfrak{u}$:

$$H(\mathfrak{W}) = -\sum_{\mathfrak{w}\in\mathrm{W}^Q} P_{\mathfrak{W}}(\mathfrak{w})\cdot\log_2(P_{\mathfrak{W}}(\mathfrak{w})) \qquad \text{(B.35)}$$

with

$$\begin{aligned}
P_{\mathfrak{W}}(\mathfrak{w}) &= \sum_{\mathfrak{u}\in\mathbb{F}_2} P_{\mathfrak{U},\mathfrak{W}}(\mathfrak{u},\mathfrak{w}) \\
&\overset{(a)}{=} \sum_{\mathfrak{u}\in\mathbb{F}_2} P_{\mathfrak{U}}(\mathfrak{u})\cdot P_{\mathfrak{W}|\mathfrak{U}}(\mathfrak{w}|\mathfrak{u}) \\
&\overset{(b)}{=} \sum_{\mathfrak{u}\in\mathbb{F}_2} P_{\mathfrak{U}}(\mathfrak{u})\cdot\prod_{q=1}^{Q} P_{\mathfrak{W}_q|\mathfrak{U}}(\mathfrak{w}_q|\mathfrak{u}) \\
&\overset{(c)}{=} \frac{1}{2}\cdot\sum_{\mathfrak{u}\in\mathbb{F}_2}\prod_{q=1}^{Q} P_{\mathfrak{W}_q|\mathfrak{U}}(\mathfrak{w}_q|\mathfrak{u}) \qquad \text{(B.36)}
\end{aligned}$$

The transition probabilities can be determined from Figure 4.22 and amount to

$$P_{\mathfrak{W}_q|\mathfrak{U}}(\mathfrak{w}_q|\mathfrak{u}) = \begin{cases} (1-P_{\mathrm{E},q})\cdot(1-P_0) & : \quad \mathfrak{w}_q = \mathfrak{u} \\ (1-P_{\mathrm{E},q})\cdot P_0 & : \quad \mathfrak{w}_q = 1-\mathfrak{u} \\ P_{\mathrm{E},q} & : \quad \mathfrak{w}_q = * \end{cases}, \quad \mathfrak{u}\in\mathbb{F}_2 \qquad \text{(B.37)}$$

The conditional entropy $H(\mathfrak{W}|\mathfrak{U})$ is finally given by

$$H(\mathfrak{W}|\mathfrak{U}) = \sum_{q=1}^{Q} H(\mathfrak{W}_q|\mathfrak{U}) \tag{B.38}$$

using the conditional independence of the channel outputs again. The conditional entropy $H(\mathfrak{W}_q|\mathfrak{U})$ of each individual subchannel can be computed by (B.30). This proofs Proposition 4.5.2.

Appendix C

Complexity of the LogMAP Algorithm

In order to derive the complexity contributions introduced by the symbol-by-symbol *Maximum A Posteriori* estimation in the *Logarithmic domain* (LogMAP) algorithm, the fundamental equations will be revisited. For a full derivation of the MAP equations in the logarithmic domain see [RVH95].

In the logarithmic domain, the forward and backward recursion $\alpha\left(\mathcal{S}_k^{(j)}\right)$ and $\beta\left(\mathcal{S}_k^{(i)}\right)$ of the well-known *Bahl-Cocke-Jelinek-Raviv* (BCJR) algorithm [BCJR74] can be expressed by means of the operator

$$\max^*(\delta_1, \delta_2) \doteq \max(\delta_1, \delta_2) + \ln\left(1 + \mathrm{e}^{-|\delta_1 - \delta_2|}\right) \tag{C.1}$$

according to

$$\alpha\left(\mathcal{S}_k^{(j)}\right) = \max^*_{\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right)} \left(\gamma\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right) + \alpha\left(\mathcal{S}_{k-1}^{(i)}\right)\right), \tag{C.2}$$

$$\beta\left(\mathcal{S}_k^{(i)}\right) = \max^*_{\left(\mathcal{S}_k^{(i)}, \mathcal{S}_{k+1}^{(j)}\right)} \left(\gamma\left(\mathcal{S}_k^{(i)}, \mathcal{S}_{k+1}^{(j)}\right) + \beta\left(\mathcal{S}_{k+1}^{(j)}\right)\right). \tag{C.3}$$

The innovation of each state transition (branch) is given by

$$\gamma^{[\mathrm{ext}]}\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right) = \frac{1}{2} \sum_{n \in \mathbb{P}} y_n L_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n), \tag{C.4}$$

$$\gamma\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right) = \gamma^{[\mathrm{ext}]}\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right) + \frac{1}{2} x_k \left(L_{\mathrm{DM}}^{[\mathrm{chan}]}(x_k) + L_{\mathrm{CD}}^{[\mathrm{apri}]}(x_k)\right), \tag{C.5}$$

exploiting the channel-related reliabilities $L_{\mathrm{DM}}^{[\mathrm{chan}]}(y_n)$ and the *a priori* information $L_{\mathrm{CD}}^{[\mathrm{apri}]}(x_k)$ regarding the input bit x_k. $\mathbb{P}$ denotes the index subset which correspond to the parity bits that contribute at the considered time instance. It is worth noting that $L_{\mathrm{CD}}^{[\mathrm{apri}]}(x_k) = 0$ in the first iteration and y_n,$x_k \in \{\pm 1\}$. The *extrinsic*

Operation	information bit	dummy bit	
		type 1	type 2
$\max^*$	$4 \cdot 2^{\mathfrak{L}} - 2$	$2 \cdot 2^{\mathfrak{L}}$	0
ADD	$8 \cdot 2^{\mathfrak{L}} + 5$	$4 \cdot 2^{\mathfrak{L}} + 2$	2

Table C.1: Complexity per information bit and dummy bit for the LogMAP channel decoder.

information is given by:

$$L_{\mathrm{CD}}^{[\mathrm{ext}]}(x_k) = \max_{\substack{\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right) \\ x_k = +1}}{}^{*} \left(\beta\left(\mathcal{S}_k^{(j)}\right) + \gamma^{[\mathrm{ext}]}\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(i)}\right) + \alpha\left(\mathcal{S}_{k-1}^{(j)}\right) \right) - \max_{\substack{\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right) \\ x_k = -1}}{}^{*} \left(\beta\left(\mathcal{S}_k^{(j)}\right) + \gamma^{[\mathrm{ext}]}\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right) + \alpha\left(\mathcal{S}_{k-1}^{(i)}\right) \right) \tag{C.6}$$

The $\max^*$ operation can be efficiently implemented using, e.g., a double lookup table containing the e-function and the ln-function. The complexity has to be evaluated separately for data bits, dummy bits for type 1 convolutional codes (type 1 dummy bits) and dummy bits for type 2 convolutional codes (type 2 dummy bits) (see Section 3.3).

Considering the complexity at data bit positions, the evaluation of (C.2) and (C.3) require $2 \cdot 2^{\mathfrak{L}}$ $\max^*$ operations and $4 \cdot 2^{\mathfrak{L}}$ additions in total. Equation (C.4) has to be evaluated only once per frame and can be reused in the iteration process, since all components are constant during decoding. Considering (C.5), $\gamma\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right)$ has to be determined for each of the $2 \cdot 2^{\mathfrak{L}}$ state transitions. However, only 4 combinations of $x_k \in \{\pm 1\}$ and $y_n \in \{\pm 1\}$ are possible resulting in 4 different values for $\gamma\left(\mathcal{S}_{k-1}^{(i)}, \mathcal{S}_k^{(j)}\right)$. Therefore, only 4 additions are required for each data bit in order to evaluate (C.5). Finally, the *extrinsic* information given in (C.6) has to be calculated by means of $(2 \cdot 2^{\mathfrak{L}} - 2)$ $\max^*$ operations and $(4 \cdot 2^{\mathfrak{L}} + 1)$ additions resulting in an overall complexity of $(4 \cdot 2^{\mathfrak{L}} - 2)$ $\max^*$ operations and $(8 \cdot 2^{\mathfrak{L}} + 5)$ additions at each data bit position.

For type 1 dummy bits, the evaluation of (C.2) and (C.3) also require $2 \cdot 2^{\mathfrak{L}}$ $\max^*$ operations and $4 \cdot 2^{\mathfrak{L}}$ additions. However, no *extrinsic* information has to be calculated and only 2 additions has to be performed in order to evaluate (C.5). The latter reduction is achieved by the constraint $x_k = 1$ at each dummy bit position which yields only two possible combination of x_k and y_n. Consequently, the total complexity is given by $2 \cdot 2^{\mathfrak{L}}$ $\max^*$ operations and $(4 \cdot 2^{\mathfrak{L}} + 2)$ additions.

Assuming a type 2 dummy bit at time instance k, only $1.5 \cdot 2^{\mathfrak{L}}$ $\max^*$ operations and $3 \cdot 2^{\mathfrak{L}}$ additions are required to evaluate (C.2) and (C.3). The reduction

for type 2 dummy bits is achieved due to the merging trellis paths. Half of the final state reliabilities $\beta\left(\mathcal{S}_k^{(i)}\right)$ can be set to zero (cf. Figure 3.9). Equation (C.5) is calculated by means of 2 additions. As explained in Section 3.3, there is also a complexity reduction achieved by type 2 dummy bits for the data bit at the succeeding time instance $k+1$. For the sake of simplicity, this reduction is incorporated into the complexity of the type 2 dummy bits at time instance k. Only half of the initial state reliabilities $\alpha\left(\mathcal{S}_k^{(j)}\right)$ have to be computed at $k+1$ yielding a reduction of $0.5 \cdot 2^{\mathfrak{L}}$ $\max^*$ operations and $2^{\mathfrak{L}}$ additions. Furthermore, the evaluation of (C.6) requires $2^{\mathfrak{L}}$ less $\max^*$ operations and $2 \cdot 2^{\mathfrak{L}}$ less additions compared to a full trellis. This results in a total complexity of $1.5 \cdot 2^{\mathfrak{L}} - 0.5 \cdot 2^{\mathfrak{L}} - 2^{\mathfrak{L}} = 0$ $\max^*$ operations and $3 \cdot 2^{\mathfrak{L}} + 2 - 2^{\mathfrak{L}} - 2 \cdot 2^{\mathfrak{L}} = 2$ additions induced by each type 2 dummy bit. The overall complexity per bit depending on its type is summarized in Table C.1.

Appendix D

Internet Protocol Suite

The internet protocol suite is a set of protocols used for the communication between adjacent layers of the TCP/IP model which is related to the ISO-OSI reference model [ISO96]. Accordingly, the protocols can either be assigned to the *application layer*, the *transport layer*, the *internet layer* or the *link layer*. Throughout this thesis, wireless and mobile communication with tight delay constraints is considered. Therefore, the following description is limited to the protocol stack consisting of the *Internet Protocol* (IP) version 4 or 6 (internet layer), the *User Datagram Protocol* (UDP) assigned to the transport layer, and the *Real-time Transport Protocol* (RTP) which is a common application layer protocol. Link layer protocols are not considered for the sake of generality, since they are highly architecture specific. Furthermore, header extensions are prohibited by the considered system to limit the packet overhead. In addition, all header fields are marked in accordance to the type of *a priori* knowledge which they provide during header decoding. It is distinguished between *known* (static) fields, fields which are *predictable* based on the information stored within the network stack, fields *without a priori knowledge* (completely unknown), and fields *with a priori knowledge* either delivered by the protocol itself or the currently available flows within the network stack.

D.1 Internet Protocol

The most prominent versions of the internet layer protocol are IP version 4 (IPv4) and IP version 6 (IPv6).

Internet Protocol version 4 (IPv4)

The IPv4 header format as specified in the RFC791 [RFC81] is depicted in Figure D.1. Its total length is 40 bytes excluding the (optional) header options field.

- **Version (V), 4 bits**: Version of IP protocol, always 4 → known.
- **Internet Header Length (IHL), 4 bits**: Length of IP header in 4 byte words, always 5 since no header options allowed → known.

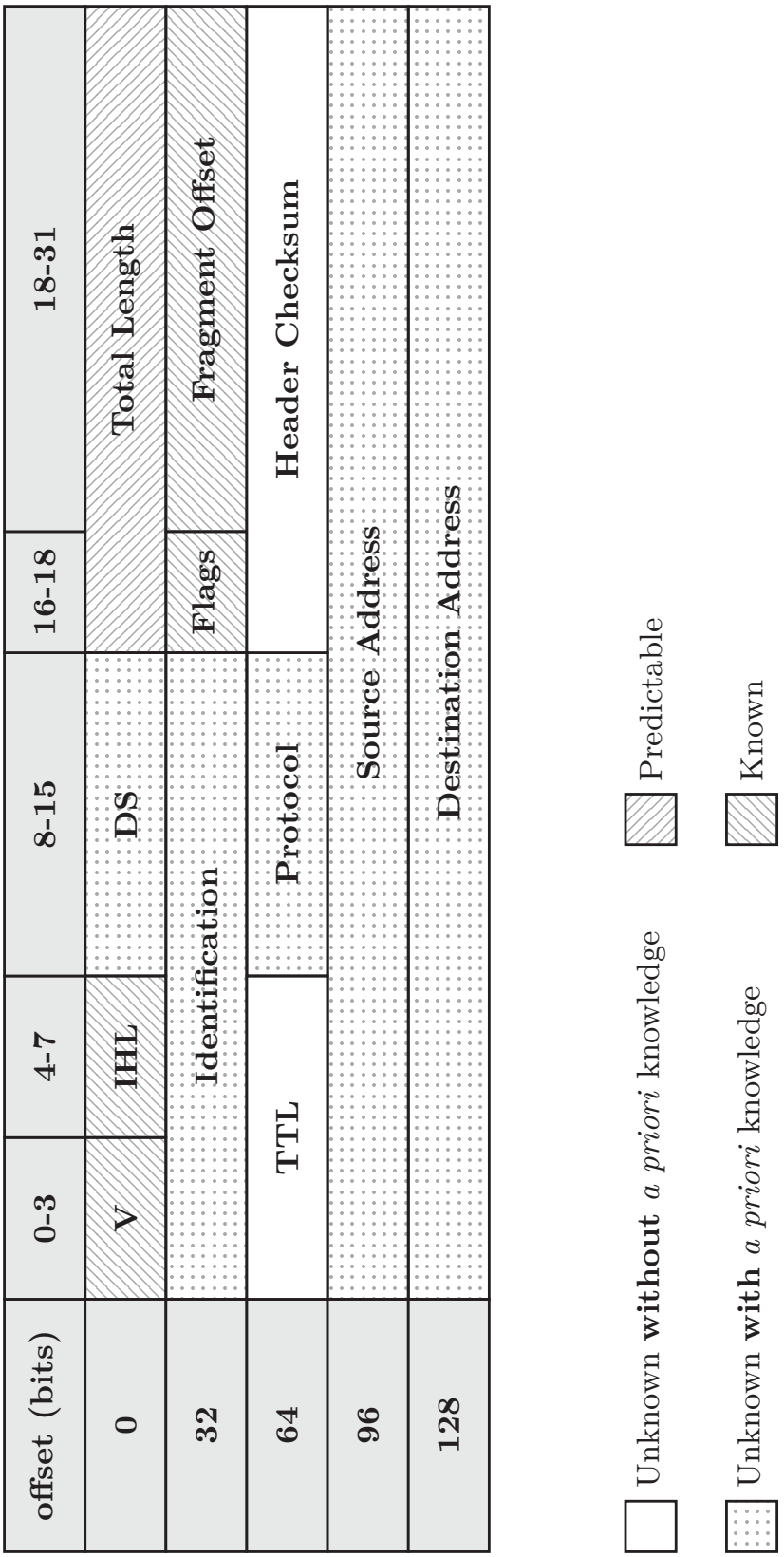

offset (bits)
0-3
4-7
8-15
16-18
18-31
0
V
IHL
DS
Total Length
32
Identification
Flags
Fragment Offset
64
TTL
Protocol
Header Checksum
96
Source Address
128
Destination Address
Unknown without a priori knowledge
Predictable
Unknown with a priori knowledge
Known

Figure D.1

- **Differentiated Services (DS), 8 bits**: Redefinition of the *Type Of Service* (TOS) field. First 6 bits define the so-called *Differentiated Service* (DS) which is required for real-time data streaming. The last 2 bits are currently unused and set to zero → unknown with protocol-related *a priori* knowledge. Furthermore, it does not change during transmission → unknown with flow-related *a priori* knowledge.
- **Total Length, 16 bits**: Length of the IP packet. If length of entire packet is known, this value can be computed → predictable.
- **Identification (ID), 16 bits**: Originally used for identifying fragments of different IP datagrams. In wireless communication systems, fragmentation is commonly carried out on MAC level and not on IP level, i.e.,ID increments by one for each packet → unknown with flow-related *a priori* knowledge.
- **Flags, 3 bits**: No fragmentation (bit sequence 010) → known.
- **Fragment Offset, 13 bits**: No fragmentation (all zero sequence) → known.
- **Time To Live (TTL), 8 bits**: Prevent too many packet cycles within the network. Each hop decrements the value by one. The packet is dropped if TTL = 0 is reached → unknown without *a priori* knowledge.
- **Protocol, 8 bits**: next encapsulated protocol → unknown with flow-related *a priori* knowledge.
- **Header Checksum, 16 bits**: A one's complement checksum of the IP header → unknown without *a priori* knowledge.
- **Source Address, 32 bits**: IP address of the sender → unknown with flow-related *a priori* knowledge.
- **Destination Address, 32 bits**: One of the IP addresses of this device → unknown with flow-related *a priori* knowledge.

Internet Protocol version 6 (IPv6)

IPv6 is a modification of IPv4 in which mainly longer IP address fields are specified. The header format is specified in the RFC2460 [RFC98] and depicted in Figure D.2. Its total length is 60 bytes.

- **Version (V), 4 bits**: Version of IP protocol, always 6 → known.
- **Traffic Class, 8 bits**: Traffic priority value, may change during transmission → unknown without *a priori* knowledge.
- **Flow Label, 20 bits**: Labels flows for which special router handling is requested → unknown with flow-related *a priori* knowledge.

offset (bits)
0-3
4-7
8-15
16-18
18-31
0
V
Traffic Class
Flow Label
32
Payload Length
Next Header
Hop Limit
64
Source Address
96
128
160
192
Destination Address
224
256
288
Unknown without a priori knowledge
Predictable
Unknown with a priori knowledge
Known

Figure D.2

- **Payload Length, 16 bits**: Length of the IP payload. If length of entire packet is known, this value can be computed → predictable.
- **Next Header, 8 bits**: next encapsulated protocol → unknown with flow-related *a priori* knowledge.
- **Hop Limit, 8 bits**: Similar to the TTL field in IPv4. Each hop decrements the value by one. The packet is dropped if TTL = 0 is reached → unknown without *a priori* knowledge.
- **Source Address, 128 bits**: IP address of the sender → unknown with flow-related *a priori* knowledge.
- **Destination Address, 128 bits**: One of the IP addresses of this device → unknown with flow-related *a priori* knowledge.

D.2 User Datagram Protocol

The User Datagram Protocol (UDP) is a transport layer protocol and mainly used for real-time and streaming applications in which the usage of the Transmission Transport Protocol (TCP) is not suitable due to delay constraints. The UDP header format of length 8 bytes is defined in RFC768 [RFC80] and shown in Figure D.3.

- **Source Port, 16 bits**: Port number of the sender (one of the open source ports) → unknown with flow-related *a priori* knowledge.
- **Destination Port, 16 bits**: Port number of the receiver (one of the open destination ports) → unknown with flow-related *a priori* knowledge.

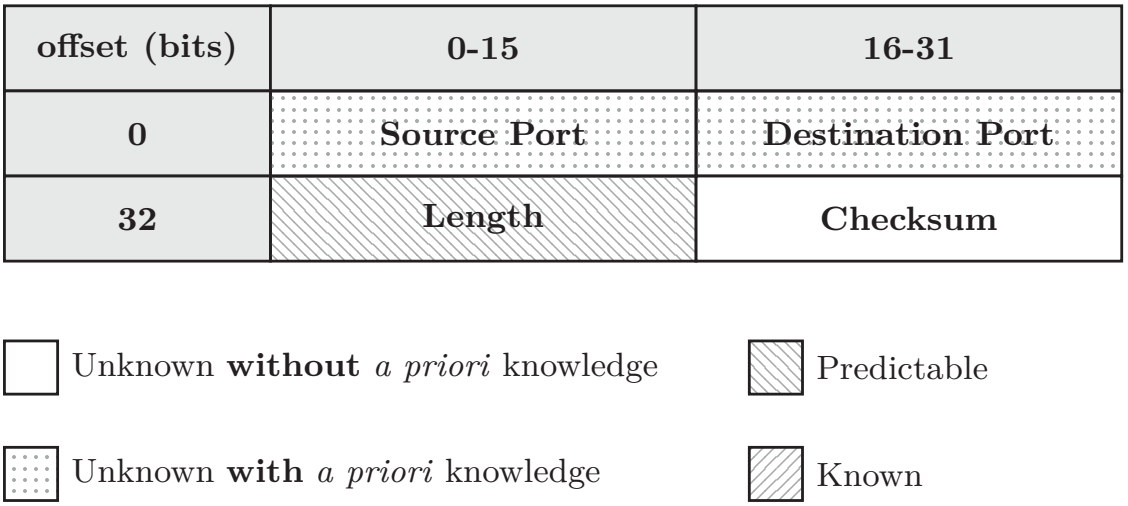


Figure D.3

- **Length, 16 bits**: Length of the UDP packet. If length of entire packet is known, this value can be computed → predictable.
- **Checksum, 16 bits**: Checksum covers IPv4 or IPv6 pseudo-header, which contain parts of the IP header and the complete UDP header → unknown without *a priori* knowledge.

D.3 Real-Time Transport Protocol

The Real-time Transport Protocol (RTP) carries application-related information and is specified in RFC1889 [RFC96]. The header format is shown in Figure D.4 (header length 12 bytes).

- **Version (V), 2 bits**: Version of RTP protocol, always 2 → known.
- **Padding (P), 1 bit**: No padding bits (header extensions not allowed) → known.
- **Extension (X), 1 bit**: X=0 (header extensions not allowed) → known.
- **CSRC Count (CC), 4 bits**: Number of contributing sources (CSRCs) which follow the fixed header. CC=0 (header extensions not allowed) → known.
- **Marker (M), 1 bits**: Interpretation application-dependent → unknown without *a priori* knowledge.
- **Payload Type (PT), 7 bits**: Format of the RTP payload, e.g, GSM audio or H263 video → unknown with flow-related *a priori* knowledge.
- **Sequence Number, 16 bits**: Sequence number increments by one for each consecutive RTP packet. Used to detect packet losses. → unknown with flow-related *a priori* knowledge (packet losses can be incorporated into the network statistic).
- **Timestamp, 32 bits**: Determines the sampling instant of the payload. Depends on the payload type → unknown with flow-related *a priori* knowledge.
- **Synchronization Source (SSRC), 32 bits**: Identifier which is chosen randomly for each synchronization source within one RTP session. Two or more sources are grouped by the receiver for playback, e.g., a microphone and a camera signal. May not change during transmission → unknown with flow-related *a priori* knowledge.

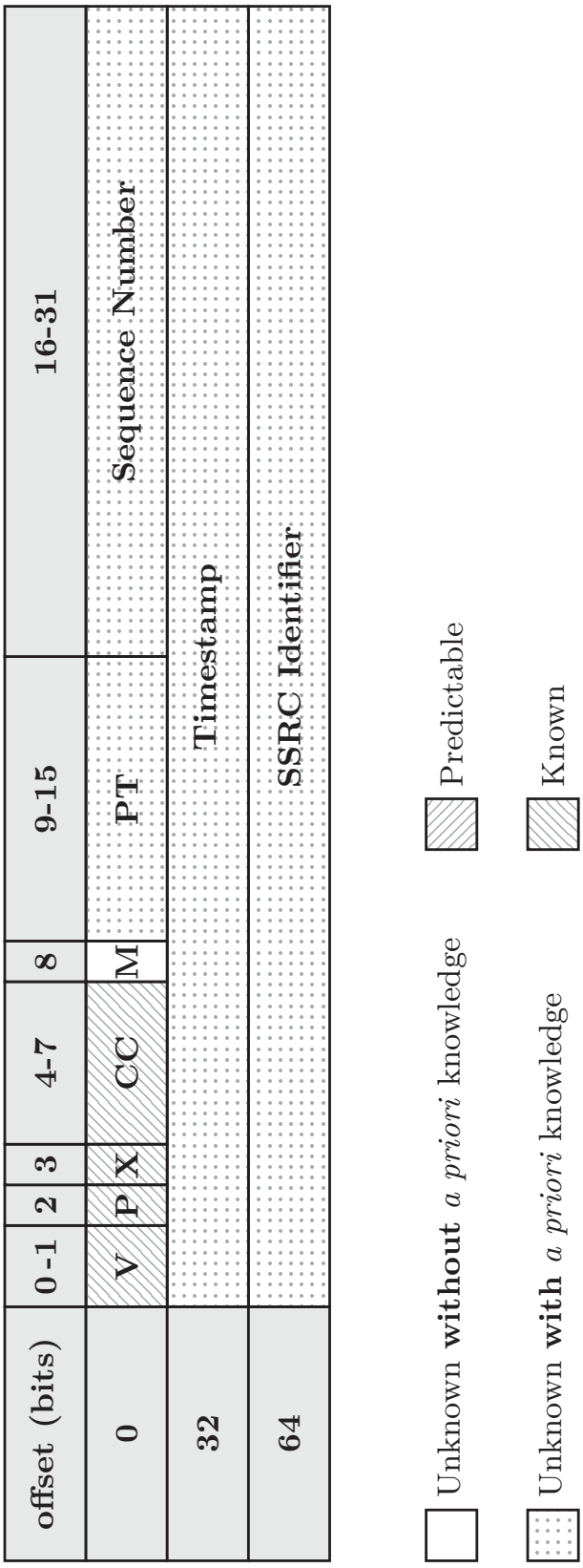

offset (bits)
0-1
2
3
4-7
8
9-15
16-31
0
V
P
X
CC
M
PT
Sequence Number
32
Timestamp
64
SSRC Identifier
Unknown without a priori knowledge
Predictable
Unknown with a priori knowledge
Known

Figure D.4

Bibliography

[3GP05] 3GPP TS 25.212. "Multiplexing and Channel Coding (FDD)". 3GPP Technical Specification Group Radio Access Network, March 2005.

[3GP06] 3GPP TS 25.309. "FDD Enhanced Uplink – General description – Stage 2". 3GPP Technical Specification Group Radio Access Network, March 2006.

[3GP08] 3GPP TR 25.943. "Deployment Aspects". 3GPP Technical Specification Group Radio Access Network, December 2008.

[3GP11] 3GPP TS 36.212. "Evolved Terrestrial Radio Access (E-UTRA); Multiplexing and Channel Coding". 3GPP Technical Specification Group Radio Access Network, April 2011.

[ACS08] M. Adrat, T. Clevorn, and L. Schmalen. "Iterative Source-Channel Decoding & Turbo DeCodulation". R. Martin, U. Heute, and C. Antweiler, editors, *Advances in Digital Speech Transmission*, chapter 13, pp. 365–398. John Wiley & Sons, Ltd, 2008.

[AKD09] U. Ali, M. Kieffer, and P. Duhamel. "Joint Protocol-Channel Decoding for Robust Aggregated Packet Recovery at WiMAX MAC Layer". *IEEE Workshop on Signal Processing Advances in Wireless Communications*, Perugia, Italy, June 2009.

[AKD10] U. Ali, M. Kieffer, and P. Duhamel. "Frame Synchronization Based on Header Recovery and Bayesian Testing". *IEEE Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Istanbul, Turkey, September 2010.

[AKD12] U. Ali, M. Kieffer, and P. Duhamel. "Joint Protocol-Channel Decoding for Robust Frame Synchronization". *IEEE Transactions on Communications (accepted for publication)*, vol. 60, no. 8, pp. 2326–2335, August 2012.

[AKtB02] A. Ashikhmin, G. Kramer, and S. ten Brink. "Extrinsic Information Transfer Functions: A Model and Two Properties". *IEEE Conference on Information Sciences and Systems (CISS)*, Princeton, New Jersey (USA), March 2002.

[AKtB04] A. Ashikhmin, G. Kramer, and S. ten Brink. "Extrinsic Information Transfer Functions: Model and Erasure Channel Properties". *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2657–2673, November 2004.

[AN07] K. Ausavapattanakun and A. Nosratinia. "Analysis of Selective-Repeat ARQ via Matrix Signal-Flow Graphs". *IEEE Transactions on Communications*, vol. 55, no. 1, pp. 198–204, January 2007.

[AOSW10] I. Aktas, J. Otten, F. Schmidt, and K. Wehrle. "Towards a Flexible and Versatile Cross-Layer-Coordination Architecture". *IEEE Conference on Computer Communications (INFOCOM)*, San Diego, CA, USA, March 2010.

[AVS01] M. Adrat, P. Vary, and J. Spittka. "Iterative Source-Channel Decoder using Extrinsic Information from Softbit-Source Decoding". *IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, Salt Lake City, UT, USA, May 2001.

[Bad09] L. Badia. "On the Effect of Feedback Errors in Markov Models for SR ARQ Packet Delays". *IEEE Global Telecommunications Confonference (GLOBECOM)*, Honolulu, HI, USA, November 2009.

[BBV11] M. Beermann, T. Breddermann, and P. Vary. "Rate-Compatible LDPC Codes using Optimized Dummy Bit Insertion". *IEEE Symposium on Wireless Communication Systems (ISWCS)*, Aachen, Germany, November 2011.

[BCH09a] N. Bonello, S. Chen, and L. Hanzo. "Pilot Symbol Assisted Coding". *Electronics Letters*, vol. 45, no. 10, pp. 518–519, May 2009.

[BCH09b] N. Bonello, S. Chen, and L. Hanzo. "On the Design of Pilot Symbol Assisted Codes". *IEEE Vehicular Technology Conference (VTC-Fall)*, Anchorage, AK, USA, September 2009.

[BCJR74] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate (Corresp.)". *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, March 1974.

[Ber71] T. Berger. *Rate-Distortion Theory*. Prentice Hall, 1971.

[BEV11] T. Breddermann, B. Eschbach, and P. Vary. "Dummy Bit Rate Matching for UMTS LTE". *ITG Fachtagung Mobilkommunikation*, Osnabrück, Germany, May 2011.

[BEV12a] T. Breddermann, B. Eschbach, and P. Vary. "Hybrid ARQ Scheme for UMTS LTE Based on Insertion Convolutional Turbo Codes". *IEEE Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sydney, Australia, September 2012.

[BEV12b] T. Breddermann, B. Eschbach, and P. Vary. "Influence of HARQ with Unreliable Feedback on the Throughput of UMTS LTE". *IEEE Vehicular Technology Conference (VTC-Fall)*, Quebec City, Canada, September 2012.

[BFC00] B. Brink, H. C. Ferreira, and W. A. Clarke. "Pruned Convolutional Codes for Flexible Unequal Error Protection Against Insertion/Deletion/Reversal Errors". *IEEE Symposium on Information Theory (ISIT)*, Sorrento, Italy, June 2000.

[BG96] C. Berrou and A. Glavieux. "Near Optimum Error Correcting Coding and Decoding: Turbo-Codes". *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, October 1996.

[BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima. "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes". *IEEE Conference on Communications (ICC)*, Geneva, Switzerland, May 1993.

[BGV05] F. Babich, GuidoMontorsi, and F. Vatta. "On Rate-Compatible Punctured Turbo Codes Design". *EURASIP Journal on Applied Signal Processing*, vol. 6, pp. 784–794, 2005.

[BLV+10] T. Breddermann, H. Lüders, P. Vary, I. Aktas, and F. Schmidt. "Iterative Source-Channel Decoding with Cross-Layer Support for Wireless VoIP". *ITG Conference on Source and Channel Coding (SCC)*, Siegen, Germany, January 2010.

[BM96a] S. Benedetto and G. Montorsi. "Serial Concatenation of Block and Convolutional Codes". *Electronics Letters*, vol. 32, no. 10, pp. 887–888, May 1996.

[BM96b] S. Benedetto and G. Montorsi. "Iterative Decoding of Serially Concatenated Convolutional Codes". *Electronics Letters*, vol. 32, no. 13, pp. 1186–1188, June 1996.

[Bou10] H. Boujemaa. "Delays Analysis of Cooperative Truncated HARQ With Opportunistic Relaying and Noisy Feedback". *IEEE Symposium on Personal Indoor and Mobile Radio Communication (PIMRC)*, Istanbul, Turkey, September 2010.

[BR05] F. Brännström and L. K. Rasmussen. "Multiple Parallel Concatenated Codes with Optimal Puncturing and Energy Distribution". *IEEE Conference on Communications (ICC)*, Soul, Korea, May 2005.

[BV10a] T. Breddermann and P. Vary. "Bad Parameter Indication for Error Concealment in Wireless Multimedia Communication". *IEEE Vehicular Technology Conference (VTC-Fall)*, Ottawa, Canada, September 2010.

[BV10b] T. Breddermann and P. Vary. "Complexity-Reduced Iterative Source-Channel Decoding Using Inner Irregular Insertion Convolutional Codes". *ITG Fachtagung Speech Communication*, Bochum, Germany, October 2010.

[BV11a] T. Breddermann and P. Vary. "EXIT Chart Optimized Rate Matching for Wireless Communication Systems". *IEEE Vehicular Technology Conference (VTC-Fall)*, San Francisco, CA, USA, September 2011.

[BV11b] T. Breddermann and P. Vary. "EXIT Functions for Parallel Concatenated Insertion Convolutional Codes". *IEEE Global Telecommunications Conference (GLOBECOM)*, Houston, TX, USA, December 2011.

[BV13] M. Beermann and P. Vary. "Breaking Cycles with Dummy Bits: Improved Rate-Compatible LDPC Codes with Short Block Lengths". *Proc. International ITG Conference on Source and Channel Coding*, Munich, Germany, January 2013.

[CBAV05] T. Clevorn, J. Brauers, M. Adrat, and P. Vary. "Turbo Decodulation: Iterative Combined Demodulation and Source-Channel Decoding". *IEEE Communications Letters*, vol. 9, no. 9, pp. 820 – 822, September 2005.

[CCG79] J. Cain, G. Clark, and J. Geist. "Punctured Convolutional Codes of Rate (n-1)/n and Simplified Maximum Likelihood Decoding". *IEEE Transactions on Information Theory*, vol. 25, no. 1, pp. 97–100, 1979.

[CCY05] Y. Choi, S. Choi, and S. Yoon. "MSDU-Based ARQ Scheme for IP-Level Performance Maximization". *IEEE Global Telecommunications Conference (GLOBECOM)*, St. Louis, MO, USA, November 2005.

[CF05] Q. Chen and P. Fan. "Performance Analysis of Hybrid ARQ with Code Combining Over Interleaved Rayleigh Fading Channel". *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 1207–1214, 2005.

[CH93] O. M. Collins and M. Hizlan. "Determinate State Convolutional Codes". *IEEE Transactions on Communications*, vol. 41, no. 12, pp. 1785–1794, 1993.

[Cha85] D. Chase. "Code Combining–A Maximum-Likelihood Decoding Approach for Combining an Arbitrary Number of Noisy Packets". *IEEE Transactions on Communications*, vol. 33, no. 5, pp. 385–393, 1985.

[Che06] J.-F. Cheng. "Coding Performance of Hybrid ARQ Schemes". *IEEE Transactions on Communications*, vol. 54, no. 6, pp. 1017–1029, 2006.

[Cis12] Cisco Visual Networking Index Mobile. "Global Mobile Data Traffic Forecast Update, 2011-2016", February 2012.

[Cle06] T. Clevorn. *Turbo DeCodulation: Iterative Joint Source-Channel Decoding and Demodulation.* PhD thesis, RWTH Aachen University, 2006.

[CMTG04] M. Conti, G. Maselli, G. Turi, and S. Giordano. "Cross-Layering in Mobile Ad Hoc Network Design". *Computer*, vol. 37, no. 2, pp. 48–51, 2004.

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* New York:Wiley, 1991.

[DK10] P. Duhamel and M. Kieffer. *Joint Source-Channel Decoding: A Cross-Layer Perspective with Applications in Video Broadcasting*, chapter 8. Elsevier Science & Technology, 2010.

[DM98] M. C. Davey and D. J. MacKay. "Low Density Parity Check Codes Over GF(q)". *IEEE Transactions on Communications*, vol. 2, no. 6, 1998.

[DP95] D. Divsalar and F. Pollara. "Multiple Turbo Codes for Deep-Space Communications". *TDA Progress Report*, pp. 42–121, May 1995.

[EBBS05] M. W. El Bahri, H. Boujemaa, and M. Siala. "Effects of Noisy Feedback on the Performance of HARQ Schemes Over Multipath Block Fading Channels for DS-SSS". *IEEE Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, vol. 4, Berlin, Germany, September 2005.

[ETS09a] ETSI. "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)". Technical report, EN 302 307 V1.21, August 2009.

[ETS09b] ETSI. “Digital Video Broadcasting (DVB); Frame structure, channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)”. Technical report, EN 302 755 V1.11, September 2009.

[FV01] T. Fingscheidt and P. Vary. “Softbit Speech Decoding: A New Approach to Error Concealment”. *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 240–251, March 2001.

[FYCL11] P. Fischer, S. Yi, S. Chun, and Y. Lee. *LTE - The UMTS Long Term Evaluation - From Theory To Practice*, chapter 4: User Plane Protocols, pp. 87–120. John Wiley & Sons, Ltd., Publication, 2 edition, 2011.

[Gal62] R. G. Gallager. “Low-Density Parity-Check Codes”. *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21 – 28, January 1962.

[Gal63] R. G. Gallager. *Low-Density Parity-Check Codes*. M.I.T. Press, Cambridge, MA, 1963.

[Gör00] N. Görtz. “Iterative Source-Channel Decoding Using Soft-In/Soft-Out Decoders”. *IEEE Symposium on Information Theory (ISIT)*, Sorrento, Italy, June 2000.

[Hag88] J. Hagenauer. “Rate-Compatible Punctured Convolutional Codes (RCPC Codes) and Their Applications”. *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 389–400, 1988.

[Hag04] J. Hagenauer. “The EXIT Chart Introduction to Extrinsic Information Transfer in Iterative Processing”. *European Signal Processing Conference (EUSIPCO)*, Vienna, Austria, September 2004.

[HEA05] X. Y. Hu, E. Eleftheriou, and D. M. Arnold. “Regular and Irregular Progressive Edge-Growth Tanner Graphs”. *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, January 2005.

[HHvD+10] W. Henkel, K. Hassan, N. von Deetzen, S. Sandberg, L. Sassatelli, and D. Declercq. “UEP Concepts in Modulation and Coding”. *Advances in Multimedia*, vol. 2010, pp. 14, 2010.

[HLY02] L. Hanzo, T. H. Liew, and B. L. Yeap. *Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels*. John Wiley & Sons, Ltd, 2002.

[HOP96] J. Hagenauer, E. Offer, and L. Papke. “Iterative Decoding of Binary Block and Convolutional Codes”. *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, 1996.

[HSS90] J. Hagenauer, N. Seshadri, and C.-E. W. Sundberg. “The Performance of Rate-Compatible Punctured Convolutional Codes for Digital Mobile Radio”. *IEEE Transactions on Communications*, vol. 38, no. 7, pp. 966–980, 1990.

[HvD06] W. Henkel and N. von Deetzen. “Path Pruning for Unequal Error Protection Turbo Codes”. *Zurich Seminar on Communications*, Zurich, Switzerland, February 2006.

[HvDH+07] W. Henkel, N. von Deetzen, K. Hassan, L. Sassatelli, and D. Declercq. “Some UEP Concepts in Coding and Physical Transport”. *IEEE Sarnoff Symposium*, Princton, NJ, USA, April 2007.

[IEE04] “IEEE 802.16e: Air Interface for Fixed and Mobile Broadband Wireless Access Systems”. IEEE Standard 802.16e, 2004.

[IEE09] “IEEE 802.11n, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 5: Enhancements for Higher Throughput”. IEEE Standard 802.11n-2009, 2009.

[ISO96] ISO/IEC 7498-1. “Information Thechnology - Open System Interconnection - Basic Reference Model: The Basic Model, Second edition”, June 1996.

[JMZD09] Z. Jaoua, A. Mokraoui-Zergaïnoh, and P. Duhamel. “Robust Transmission of 802.11N Physical Packet Headers”. *IEEE Workshop on Signal Processing Advances in Wireless Communications*, Perugia, Italy, June 2009.

[JSX05] H. Jenkač, T. Stockhammer, and W. Xu. “Permeable-Layer Receiver for Reliable Multicast Transmission in Wireless Systems”. *IEEE Conference on Wireless Communications and Networking*, New Orleans, LA, USA, March 2005.

[Kal90] S. Kallel. “Analysis of a Type II Hybrid ARQ Scheme With Code Combining”. *IEEE Transactions on Communications*, vol. 38, no. 8, pp. 1133–1137, 1990.

[KD10] M. Kieffer and P. Duhamel. “Joint Protocol and Channel Decoding: An Overview”. *Future Network and Mobile Summit*, Florence, Italy, June 2010.

[KGHL05] H. M. Kwon, A. Goyal, K. Hassan, and Y. H. Lee. "Encoded Pilot Placement for LDPC Iterative Receiver". *IEEE Vehicular Technology Conference (VTC-Fall)*, Dallas, TX, USA, September 2005.

[KH90] S. Kallel and D. Haccoun. "Generalized Type II Hybrid ARQ Scheme Using Punctured Convolutional Coding". *IEEE Transactions on Communications*, vol. 38, no. 8, pp. 1938–1946, 1990.

[KH08] A. A. Kadhim and A. A. Hamad. "Turbo Codes with Internal Pilot Insertion". *IEEE Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST)*, Cardiff, Wales, UK, September 2008.

[KHG+05] H. M. Kwon, K. Hassan, A. Goyal, M.-K. Oh, D.-J. Park, and Y. H. Lee. "Encoded Pilots for Iterative Receiver Improvement". *IEEE Vehicular Technology Conference (VTC-Fall)*, Dallas, TX, USA, September 2005.

[KMFS03] O. Kato, A. Matsumoto, K. Fukawa, and H. Suzuki. "Packet Error Rate Analysis and its Reduction by Known Bits Insertion for Turbo Code in 100Mbps OFCDM System". *IEEE Vehicular Technology Conference (VTC-Spring)*, Vancouver, Canada, September 2003.

[LC82] S. Lin and D. J. Costello. *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, 1982.

[LH06] I. Land and J. Huber. "Information Combining". *Foundations and Trends in Communications and Information Theory*, vol. 3, no. 3, 2006.

[LHHH05] I. Land, S. Huettinger, P. A. Hoeher, and J. B. Huber. "Bounds on Information Combining". *IEEE Transactions on Information Theory*, vol. 51, no. 2, pp. 612–619, 2005.

[LLMC12] C. J. Le Martret, A. Le Duc, S. Marcille, and P. Ciblat. "Analytical Performance Derivation of Hybrid ARQ Schemes at IP Layer". *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1305–1314, May 2012.

[LMS07] C. Lott, O. Milenkovic, and E. Soljanin. "Hybrid ARQ: Theory, State of the Art and Future Directions". *IEEE Information Theory Workshop on Information Theory for Wireless Networks*, Bergen, Norway, July 2007.

[LMSS01] M. G. Luby, M. Miteznmacher, M. A. Shokrollahi, and D. A. Spielman. "Improved Low-Density Parity-Check Codes Using Irregular

Graphs". *IEEE Journal on Information Theory*, vol. 47, no. 2, pp. 585–598, 2001.

[LN02] J. Li and K. R. Narayanan. "Rate-Compatible Low Density Parity Check Codes for Capacity-Approaching ARQ Schemes in Packet Data Communication". *IEEE Conference on Communications, Internet and Information Technology (CIIT)*, Virgin Islands, USA, November 2002.

[LPRP99] J. Lee, K. Park, K. Ryu, and P. Park. "The Optimum Number of Inserted Known-Bits in the Rate Matched Turbo Codes with Known Bits". *IEEE Region 10 Conference TENCON 99*, vol. 1, pp. 12–14, 1999.

[LR97] X. Li and J. A. Ritcey. "Bit-Interleaved Coded Modulation with Iterative Decoding". *IEEE Communications Letters*, vol. 1, no. 6, pp. 169–171, November 1997.

[LS00] Z. Lin and A. Svensson. "New Rate-Compatible Repetition Convolutional Codes". *IEEE Transactions on Information Theory*, vol. 46, no. 7, pp. 2651–2659, 2000.

[Lüd10] H. E. Lüders. *Current and Evolved Physical Layer Concepts: Potentials and Limitations of Mobile Broadband Wireless Access.* PhD thesis, RWTH Aachen University, 2010.

[LWZ09] X. Liu, X. Wu, and C. Zhao. "Shortening for Irregular QC-LDPC Codes". *IEEE Communications Letters*, vol. 13, no. 8, pp. 612–614, August 2009.

[Mac97] D. J. C. MacKay. "Near Shannon Limit Performance of Low Density Parity Check Codes". *Electronics Letters*, vol. 33, no. 6, pp. 457–458, March 1997.

[Mac99] D. J. C. MacKay. "Good Error-Correcting Codes Based on Very Sparse Matrices". *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, March 1999.

[Man74] D. Mandelbaum. "An Adaptive-Feedback Coding Scheme Using Incremental Redundancy". *IEEE Transactions on Information Theory*, vol. 20, no. 3, pp. 388–389, 1974.

[MCM11] S. Marcille, P. Ciblat, and C. J. L. Martret. "Performance Computation of Cross-Layer Hybrid ARQ: Schemes at IP Layer in the Presence of Corrupted Acknowledgments". *IEEE International Workshop on Cross-Layer Design (IWCLD)*, Rennes, France, November 2011.

[MK99] R. Mantha and F. R. Kschischang. "A Capacity-Approaching Hybrid ARQ Scheme Using Turbo Codes". *IEEE Global Telecommunications Conference (GLOBECOM)*, Rio De Janeiro, Brazil, December 1999.

[MK09] A. Mukherjee and H. M. Kwon. "CSI-Adaptive Encoded Pilot-Symbols for Iterative OFDM Receiver with IRA Coding". *IEEE Vehicular Technology Conference (VTC-Spring)*, Barcelona, Spain, April 2009.

[MK10] F. Mériaux and M. Kieffer. "Robust IP and UDP-Lite Header Recovery for Packetized Multimedia Transmission". *IEEE Conference on Acoustics Speech and Signal Processing (ICASSP)*, Dallas, TX, USA, March 2010.

[ML00] E. Malkamaki and H. Leib. "Performance of Truncated Type-II Hybrid ARQ Schemes With Noisy Feedback Over Block Fading Channels". *IEEE Transactions on Communications*, vol. 48, no. 9, pp. 1477–1487, 2000.

[MLKD08a] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel. "Robust Header Recovery Based Enhanced Permeable Protocol Layer Mechanism". *IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Recife, Brazil, July 2008.

[MLKD08b] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel. "Robust MAC-Lite and Header Recovery Based Improved Permeable Protocol Layer Scheme". *IEEE International Symposium on Spread Spectrum Techniques and Applications (ISSSTA)*, Bologna, Italy, August 2008.

[MLKD10] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel. "Robust MAC-Lite and Soft Header Recovery for Packetized Multimedia Transmission". *IEEE Transactions on Communications*, vol. 58, no. 3, pp. 775–784, 2010.

[Moo05] T. K. Moon. *Error Correction Coding, Mathematical Methods and Algorithms*. Wiley, 2005.

[MS01] B. Mielczarek and A. Svensson. "joint Adaptive Rate Turbo Decoding and Synchronization on Rayleigh Fading Channels". *IEEE Vehicular Technology Conference (VTC-Spring)*, June 2001.

[MWD99] D. J. C. MacKay, S. T. Wilson, and M. C. Davey. "Comparison of Constructions of Irregular Gallager Codes". *IEEE Transactions on Communications*, vol. 47, no. 10, pp. 1449–1454, 1999.

[Oka08] T. Okamura. "A Hybrid ARQ Scheme Based on Shortened Low-Density Parity-Check Codes". *IEEE Wireless Communications and Networking Conference (WCNC)*, Las Vegas, NV, USA, March 2008.

[OKPL08] M.-K. Oh, H. M. Kwon, D.-J. Park, and Y. H. Lee. "Iterative Channel Estimation and LDPC Decoding With Encoded Pilots". *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 273–285, 2008.

[OS97] M. Öberg and P. H. Siegel. "Application of Distance Spectrum Analysis to Turbo Code Performance Improvement". *Allerton Conference on Communication, Control and Computing*, Urbana-Champaign, IL, USA, 1997.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, USA, 1988.

[Per07] F. Persson. "Voice over IP Realized for the 3GPP Long Term Evolution". *IEEE Vehicular Technology Conference (VTC-Fall)*, Baltimore, MD, USA, September 2007.

[RFC80] "RFC 768: User Datagram Protocol", August 1980.

[RFC81] "RFC 791: Internet Protocol – Darpa Internet Program Protocol Specification", September 1981.

[RFC96] "RFC 1889: RTP: A Transport Protocol for Real-Time Applications", January 1996.

[RFC98] "RFC 2460: Internet Protocol, Version 6 (IPv6) Specification", December 1998.

[RI04] V. Raisinghani and S. Iyer. "ECLAIR: An Efficient Cross Layer Architecture for Wireless Protocol Stacks". *World Wireless Congress (WWC)*, San Francisco, CA, USA, May 2004.

[RM00] D. N. Rowitch and L. B. Milstein. "On the Performance of Hybrid FEC/ARQ Systems Using Rate Compatible Punctured Turbo (RCPT) Codes". *IEEE Transactions on Communications*, vol. 48, no. 6, pp. 948–959, 2000.

[RSU01] T. Richardson, M. Shokrollahi, and R. Urbanke. "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes". *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, February 2001.

[RU01a] T. J. Richardson and R. Urbanke. "The Capacity of Low-Density Parity-Check Codes under Message-Passing Decoding". *IEEE Transactions on Information Theory*, vol. 47, February 2001.

[RU01b] T. J. Richardson and R. L. Urbanke. "Efficient Encoding of Low-Density Parity-Check Codes". *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 638–658, 2001.

[RU08] T. Richardson and R. Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008.

[RVH95] P. Robertson, E. Villebrun, and P. Hoeher. "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithm Operating in the Log Domain". *IEEE International Conference on Communications (ICC)*, Seattle, WA, USA, June 1995.

[RZ08] C. Ruland and N. Zivic. "Feedback in Joint Channel Coding and Cryptography". *ITG Conference on Source and Channel Coding (SCC)*, Ulm, Germany, January 2008.

[SAAW11] F. Schmidt, M. H. Alizai, I. Aktaş, and K. Wehrle. "Refector: Heuristic Header Error Recovery for Error-Tolerant Transmissions". *ACM conference on emerging Networking EXperiments and Technologies (CoNEXT)*, Tokyo, Japan, December 2011.

[SCV04] S. Sesia, G. Caire, and G. Vivier. "Incremental Redundancy Hybrid ARQ Schemes Based on Low-Density Parity-Check Codes". *IEEE Transactions on Communications*, vol. 52, no. 8, pp. 1311–1321, 2004.

[Sha48] C. E. Shannon. "A Mathematical Theory of Communication". *The Bell System Technical Journal*, vol. 27, pp. 379 – 423, 623 – 656, July and October 1948.

[SM09] R. Susitaival and M. Meyer. "LTE Coverage Improvement by TTI Bundling". *IEEE Vehicular Technology Conference (VTC-Spring)*, Barcelona, Spain, April 2009.

[SM12] S. M.-S. Sadough and M. Modarresi. "An Improved Rate Matching Method for DVB Systems Through Pilot Bit Insertion". *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 18, pp. 3251–3256, 2012.

[STBV11] L. Schmalen, M. Tschauner, T. Breddermann, and P. Vary. "Novel Iterative Multiple Description Coding for Correlated Sources". *Proc. European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, August 2011.

[Tan81] R. M. Tanner. “A Recursive Approach to Low Complexity Codes”. *IEEE Transactions on Informations Theory*, vol. 27, no. 5, pp. 533–547, September 1981.

[tB00a] S. ten Brink. “Design of Serially Concatenated Codes Based on Iterative Decoding Convergence”. *IEEE International Symposium on Turbo Codes & Related Topics (ISTC)*, Brest, France, September 2000.

[tB00b] S. ten Brink. “Iterative Decoding Trajectories of Parallel Concatenated Codes”. *ITG Conference on Source and Channel Coding (SCC)*, Munich, Germany, January 2000.

[tB01a] S. ten Brink. “Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes”. *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.

[tB01b] S. ten Brink. “Exploiting the Chain Rule of Mutual Information for the Design of Iterative Decoding Schemes”. *Allerton Conference on Communication, Control and Computing*, Urbana-Champaign, IL, USA„ October 2001.

[Tho07] R. Thobaben. “EXIT Functions for Randomly Punctured Systematic Codes”. *IEEE Information Theory Workshop (ITW)*, Lake Tahoe, CA, USA, September 2007.

[TJ05] T. Tian and C. R. Jones. “Construction of Rate-Compatible LDPC Codes Utilizing Information Shortening and Parity Puncturing”. *EURASIP Journal on Wireless Communications and Networking*, vol. 5, pp. 789 – 795, 2005.

[WC98] C.-H. Wang and C.-C. Chao. “Path-Compatible Pruned Convolutional (PCPC) Codes: A New Scheme for Unequal Error Protection”. *IEEE International Symposium on Information Theory (ISIT)*, Cambridge, MA, USA, August 1998.

[WC02] C.-H. Wang and C.-C. Chao. “Path-Compatible Pruned Convolutional (PCPC) Codes”. *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 213–224, 2002.

[WJ09] M. Westerlund and I. Johansson. “RFC 5404: RTP Payload Format for G.719”, January 2009.

[WL08] C.-H. Wang and Y.-H. Lin. “Combined Puncturing and Path Pruning for Convolutional Codes and the Application to Unequal Error Protection”. *IEEE Transactions on Communications*, vol. 56, no. 9, pp. 1385–1389, 2008.

[Wu10] P. Wu. "Performance of Hybrid-ARQ in Block-Fading Channels: A Fixed Outage Probability Analysis". *IEEE Transactions on Communications*, vol. 58, no. 4, pp. 1129–1141, April 2010.

[XM01] W. Xu and M. Marke. "Using Insertion Convolutional Codes for Speech Transmission Over GERAN 8PSK Voice Bearers". *IEEE Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, San Diego, CA, USA, September 2001.

[XR00] W. Xu and J. Romme. "A Class of Multirate Convolutional Codes by Dummy Bit Insertion". *IEEE Global Telecommunications Conference (GLOBECOM)*, San Francisco, CA, USA, November 2000.

[Ziv08] N. Zivic. "Impact of Soft Input Decryption Using Feedback on Channel Decoding". *Conference on Innovations in Information Technology (IIT)*, Monterey, CA, USA, June 2008.

[ZR95a] M. Zorzi and R. R. Rao. "Throughput Analysis of Go-Back-N ARQ in Markov Channels with Unreliable Feedback". *IEEE International Conference on Communications (ICC)*, Seattle, WA, USA, June 1995.

[ZR95b] M. Zorzi and R. R. Rao. "Performance of ARQ Go-Back-N Protocol in Markov Channels With Unreliable Feedback: Delay Analysis". *IEEE Conference on Universal Personal Communications*, Tokyo, Japan, November 1995.

[ZRR09a] N. Zivic, O. U. Rehman, and C. Ruland. "Analysis of Serial And Parallel Soft Input Decryption Schemes Over a Wireless Channel". *IEEE Symposium on Performance Evaluation of Computer & Telecommunication Systems (SPECTS)*, Istanbul, Turkey, July 2009.

[ZRR09b] N. Zivic, C. Ruland, and O. U. Rehman. "Error Correction Over Wireless Channels Using Symmetric Cryptography". *Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)*, Aalborg, Denmark, May 2009.

[ZS05] X. Zhaoji and B. Sebire. "Impact of ACK/NACK Signalling Errors on High Speed Uplink Packet Access (HSUPA)". *IEEE International Conference on Communications (ICC)*, Seoul, Korea, May 2005.

[ZT08] N. Zivic and S. Tcaciuc. "Feedback Using Dummy Bits". *Applied Computing Conference*, Istanbul, Turkey, May 2008.