Rateless Coding in the Finite Length Regime

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

vorgelegt von

Diplom-Ingenieurin

Birgit Elke Schotsch

aus Victoria

Berichter: Universitätsprofessor Dr.-Ing. Peter Vary
Universitätsprofessor Dr.-Ing. habil. Johannes B. Huber

Tag der mündlichen Prüfung: 18. Juli 2014

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.

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 38)
ISSN 1437-6768
ISBN 978-3-86073-837-5

Wissenschaftsverlag Mainz
Süsterfeldstr. 83, 52072 Aachen
Tel.: 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
www.druckereimainz.de

Gedruckt auf chlorfrei gebleichtem Papier

D 82 (Diss. RWTH Aachen University, 2014)

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Institut für Nachrichtengeräte und Datenverarbeitung (IND) der Rheinisch-Westfälischen Technischen Hochschule Aachen im Rahmen des DFG Forschungsclusters UMIC (Ultra high-speed Mobile Information and Communication). Ich möchte an dieser Stelle gerne die Gelegenheit ergreifen, all jenen zu danken, die mich bei diesem Unterfangen begleitet und unterstützt haben.

Mein besonders herzlicher Dank gilt Herrn Prof. Dr.-Ing. Peter Vary für die Betreuung dieser Dissertation, die wertvollen Anregungen und seine uneingeschränkte Unterstützung. Auch bei Herrn Prof. Dr.-Ing. habil. Johannes B. Huber bedanke ich mich sehr herzlich für sein großes Interesse an meiner Arbeit, die intensiven Diskussionen und für die Übernahme des Koreferats.

Allen Kolleginnen und Kollegen danke ich für die wunderbare Zusammenarbeit und die freundschaftliche Atmosphäre am IND. Ganz besonders möchte ich jedoch diejenigen Kollegen hervorheben, die durch fachliche Diskussionen, Anregungen oder Korrekturlesen zu dieser Arbeit beigetragen haben: Herr Dr.-Ing. Bernd Geiser, Herr Dipl.-Ing. Moritz Beermann, Herr Tim Schmitz, M.Sc. und Herr Dipl.-Ing. Benedikt Eschbach. Des Weiteren bedanke ich mich auch bei Herrn Dipl.-Ing. Henning Schepker, Herrn Radu Lupoaie, M.Sc., Herrn Dipl.-Ing. Wilfried Winkelhüsener und Frau Xin Yuan, M.Sc., die mit ihren Diplom- bzw. Masterarbeiten wichtige Beiträge geleistet haben.

Schließlich danke ich ganz besonders meinen Eltern, Hildegard und Wilhelm Schotsch, die mir das Studium der Elektrotechnik und Informationstechnik ermöglicht haben und mich stets nach Kräften unterstützt haben. Lieber Andreas, Dir danke ich von ganzem Herzen für Dein Verständnis, Deinen Zuspruch und Deine Liebe.

Aachen, im September 2014 *Birgit Schotsch*

Abstract

Rateless codes, also known as digital fountain codes, are excellently suited for erasure correction in packet-switched communication networks. First applications are digital video broadcast or multicast over terrestrial networks or multimedia services in cellular networks. Such networks are usually prone to packet losses due to network congestions or unrecoverable bit errors within packets. The main attributes of rateless codes can be summarised as follows:

- The transmitter is able to produce as many encoded packets as needed from a given source block consisting of k source packets.
- The receiver is able to decode an exact copy of the entire source block from any subset of $k(1+\varepsilon_{\mathrm{R}})$ received (i.e. non-erased) encoded packets, where $\varepsilon_{\mathrm{R}} \geq 0$ is a small reception overhead.
- No feedback channel is required for packet acknowledgements.

In the literature, rateless codes are usually based on the simplifying design assumptions of input sequences of infinite length. The analysis and the characterisation of the so designed codes apply only to codes with very long input sequences and a corresponding latency. In contrast, this thesis focuses codes with finite (especially short to medium) lengths. These practical lengths enable applications that require a low transmission latency. In this context, various types of finite length LT codes and Raptor codes are investigated. The main contributions of this thesis are:

- The derivation of analytical closed form expressions of the residual erasure probability under optimal decoding.
- The derivation of tight upper and lower residual erasure bounds.
- The generalisation of binary codes to higher order Galois fields.
- The formulation of concrete design guidelines for highly efficient LT code ensembles with equal and unequal erasure protection.
- New performance assessment tools for Raptor codes in terms of the so-called erasure weight and kernel weight profiles.

The key to the achieved results is to formulate the expected erasure correction performance of an LT code ensemble as an equivalent mathematical problem. This fundamental question is whether a consistent system of designed random linear equations over a finite field can be solved partially or completely.

Contents

Notation and Symbols

The following notation is used throughout this thesis to denote different quantities: Scalars are written in italic type (e.g. x). Boldfaced lower case letters denote *column* vectors (e.g. $\mathbf{x}$), while boldfaced capital letters denote matrices (e.g. $\mathbf{X}$). The corresponding random variables are set in sans serif font, e.g. x for random variables, $\boldsymbol{\mathsf{x}}$ for random vectors and $\boldsymbol{\mathsf{X}}$ for random matrices. Sets are written in calligraphic letters (e.g. $\mathcal{X}$). The indices "T" and "R" are used to differentiate between transmitter and receiver related quantities (e.g. $\mathbf{x}_\mathrm{T}$ and $\mathbf{x}_\mathrm{R}$). Quantities that are stated in a general manner or which are equally related to both transmitter and receiver are written without an index "T" or "R".

List of Principal Symbols

$a(\xi)$	Primitive polynomial of a Galois field $\mathbb{F}_q$ with coefficients a_i from the prime subfield $\mathbb{F}_2$
α	Primitive element of a Galois field $\mathbb{F}_q$
$\mathbf{A}$	Companion matrix to the primitive element $\alpha \in \mathbb{F}_q$
$\mathcal{A}$	Ambient space
$\mathcal{A}^\star$	Minimal ambient space
$\mathbf{b}$	Base vector
β	Complexity factor
$\mathfrak{C}$	Stochastic parity-check ensemble
d	Output node degree or row weight
d_τ	Row weight in that part of the LT code generator matrix which is associated with importance class τ
$\mathbf{d}$	Vector of importance class row weights, i.e. $\mathbf{d} = (d_1, \ldots, d_T)^\mathsf{T}$
d	Output node degree or row weight (random variable)
d_τ	Row weight (random variable) in that part of the LT code generator matrix which is associated with importance class τ
$\boldsymbol{\mathsf{d}}$	Vector of importance class row weights (random vector)

$\mathcal{D}$	Row weight sample space, i.e. set of row weights with non-zero probabilities
$\delta_{\mathrm{H,min}}$	Minimum Hamming distance
Δ	Density of a matrix, i.e. relative amount of non-zero entries
ε	Relative overhead
ε_{R}	Relative reception overhead
ε_{T}	Relative transmission overhead
η_{R}	Absolute symbol reception overhead, $\eta_{\mathrm{R}} = k\varepsilon_{\mathrm{R}} = k(\gamma_{\mathrm{R}} - 1)$
ϵ	Erasure probability on the BEC or the SEC
$\mathbb{F}_2$	Galois field of order 2
$\mathbb{F}_q$	Galois field of order q
g_{ij}	An entry in the i^{th} row and j^{th} column of an LT code generator matrix
g_{ij}	An entry (random variable) in the i^{th} row and j^{th} column of an LT code generator matrix
$\mathbf{G}$	LT code generator matrix
$\boldsymbol{\mathsf{G}}$	LT code generator matrix (random matrix)
$\mathcal{G}$	Sample space of matrices $\mathbf{G}$
γ	Inverse code rate
γ_{R}	Inverse reception code rate
γ_{T}	Inverse transmission code rate
$\mathbf{H}$	Parity-check matrix
$\boldsymbol{\mathsf{H}}$	Random parity-check matrix
$\mathfrak{H}$	Hamming code
$\mathfrak{H}_{\sigma}$	Shortened Hamming code
$\mathfrak{H}_{q,\sigma}$	Shortened non-binary Hamming code
$\mathfrak{H}^{+}$	Extended Hamming code
$\mathfrak{H}^{+}$	Shortened extended Hamming code
$\mathbf{I}_{i\times i}$	Identity matrix of size $i \times i$
k	Number of symbols per information word. Using Raptor codes, k denotes the number of intermediate symbols and k' denotes the number of symbols per information word.
k_{B}	Number of bits per information word, i.e. $k_{\mathrm{B}} = \mu k$

k_τ	Number of input symbols in importance class τ; In Chapter 4, k_1 and k_2 denote the number of systematically and non-systematically received input symbols, respectively.
$\mathbf{k}$	Vector containing the numbers of input symbols from $\mathbb{F}_q$ that are assigned to T importance classes, i.e. $\mathbf{k} = (k_1, \ldots, k_T)^\mathsf{T}$
$\mathfrak{L}$	LT code ensemble
$\Lambda(\xi)$	Variable node degree distribution of an LDPC code
Λ_i	i^{th} coefficient of the variable node degree distribution of an LDPC code
m	Number of parity bits or symbols
μ	Number of bits per $\mathbb{F}_q$-element, i.e. $\mu = \mathrm{ld}(q)$
n	Codeword length
n_R	Number of received output symbols or nodes
n_T	Number of transmitted output symbols or nodes
N	Cardinality of a set or number of possible combinatorial outcomes
$\mathbb{N}$	The set of natural numbers
$\mathbb{N}_0$	The set of natural numbers including zero
o	Nullity of a matrix
$\Omega(\xi)$	Row weight or output node degree distribution of an LT code
$\boldsymbol{\Omega}(\boldsymbol{\xi})$	Multivariate row weight distribution of an LT code
Ω_i	i^{th} coefficient of the row weight distribution $\Omega(\xi)$
$\Omega_\mathbf{d}$	Coefficient of the multivariate row weight distribution $\boldsymbol{\Omega}(\boldsymbol{\xi})$
$\overline{P}^{[\mathfrak{L}]}\left(\cancel{W}\right)$	An upper bound on the residual word erasure probability
$P^{[\mathfrak{L}]}\left(\cancel{W}\right)$	Residual word erasure probability of an LT code ensemble
$\underline{P}^{[\mathfrak{L}]}\left(\cancel{W}\right)$	A lower bound on the residual word erasure probability
$\overline{P}^{[\mathfrak{L}]}\left(\cancel{S}\right)$	An upper bound on the residual symbol erasure probability
$P^{[\mathfrak{L}]}\left(\cancel{S}\right)$	Residual symbol erasure probability of an LT code ensemble
$\underline{P}^{[\mathfrak{L}]}\left(\cancel{S}\right)$	A lower bound on the residual symbol erasure probability
$\mathfrak{P}$	Precode
φ	Weighting factor
$\boldsymbol{\varphi}$	Vector of weighting factors or odds
q	Order of a Galois field $\mathbb{F}_q$
r	Rank of a matrix

$\mathbf{r}^\mathsf{T}$	Row vector from an LT code generator matrix
$\boldsymbol{\mathsf{r}}^\mathsf{T}$	Row vector from an LT code generator matrix (random vector)
$R(\xi)$	Check node degree distribution of an LDPC code
R_i	i^th coefficient of the check node degree distribution of an LDPC code
ρ	Code rate
ρ_R	Reception code rate
ρ_T	Transmission code rate
$\mathbf{s}$	Syndrome
σ	Shortening parameter, used for shortening Hamming codes
τ	Importance class index
T	Number of importance classes
$\boldsymbol{\mathsf{v}}$	A random vector that results as the element-wise product of an input vector $\mathbf{x}$ and a row $\boldsymbol{\mathsf{r}}$ from the random LT matrix $\boldsymbol{\mathsf{G}}$
w_e	Erasure weight
w_e	Erasure weight (random variable)
w_k	Kernel weight
w_k	Kernel weight (random variable)
x	Message/input symbol/node
$\mathbf{x}$	Vector of message/input symbols or nodes, input vector
ξ	Indeterminate
$\Xi(\xi)$	Input node degree distribution of an LT code
Ξ_i	i^th coefficient of the input node degree distribution $\Xi(\xi)$
y	Output/encoded bit/symbol/node
$\mathbf{y}$	Vector of output/encoded bits/symbols/nodes
$?$	Erasure symbol

Vectors, Matrices and Ensembles

$\mathbf{x}$	Bold lower case letters are understood as column vectors
$\boldsymbol{\mathsf{x}}$	Bold lower case sans serif letters denote random column vectors
$\mathbf{X}$	Bold capital letters represent matrices
$\boldsymbol{\mathsf{X}}$	Bold capital sans serif letters represent random matrices
$\mathbf{x}^\mathsf{T},\ \mathbf{X}^\mathsf{T}$	Transposition of a vector or a matrix

$\mathbf{X} \sim \Omega(\xi)$	The Hamming weight of the row vectors of random matrix $\mathbf{X}$ is distributed according to $\Omega(\xi)$

Operators

$i!$	Factorial of a non-negative integer i, i.e. $i! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot (i-1) \cdot i$
$\binom{i}{j}$	Binomial coefficient, $\binom{i}{j} = \frac{i!}{j!(i-j)!} > 0$ if $i,\, j \in \mathbb{N}_0$ and $0 \leq j \leq i$. In all other cases $\binom{i}{j} = 0$ applies.
$[i]_q$	q-analogue, q-number or q-bracket of an integer i, $[i]_q = \frac{q^i - 1}{q-1}$
$[i]_q!$	q-factorial, i.e. $[i]_q! = [1]_q \cdot [2]_q \cdot \ldots \cdot [i-1]_q \cdot [i]_q$
$\begin{bmatrix} i \\ j \end{bmatrix}_q$	Gaussian or q-binomial coefficient, i.e. $\begin{bmatrix} i \\ j \end{bmatrix}_q = \frac{[i]_q!}{[i-j]_q! [j]_q!}$
$\lceil x \rceil$	Smallest integer larger than or equal to x
$\lvert \mathcal{X} \rvert$	Cardinality of a set $\mathcal{X}$, i.e. the number of elements in $\mathcal{X}$
$\lvert x \rvert$	Absolute value of x
$\lVert \mathbf{x} \rVert_{\mathrm{H}}$	Hamming weight operator or zero "norm": counts the number of non-zero elements in vector $\mathbf{x}$
$\lVert \mathbf{x} \rVert_\infty$	Infinity norm, i.e. $\lVert \mathbf{x} \rVert_\infty = \max(\lvert x_1 \rvert,\, \lvert x_2 \rvert, \ldots)$
δ_i	Kronecker delta function, $\delta_i = 1$ if $i = 0$ and $\delta_i = 0$ if $i \neq 0$
$\delta_{i,j} = \delta_{i-j}$	Kronecker delta function, $\delta_{i,j} = 1$ if $i = j$ and $\delta_{i,j} = 0$ if $i \neq j$
$\mathrm{coef}\big(X(\xi),\, \xi^i\big)$	coefficient of ξ^i in a polynomial $X(\xi)$
$\dim(\mathbf{x_1},\, \mathbf{x_2}, \ldots)$	Dimension of the subspace spanned by a set of vectors $\mathbf{x_1},\, \mathbf{x_2}, \ldots$
$\mathrm{ld}(.)$	Logarithm with a basis of 2
$i \mod j$	The modulo function computes the remainder of the division i/j
$\mathrm{rnd}(.)$	Rounding operator
$\mathrm{img}(\mathbf{X})$	Image of a matrix $\mathbf{X}$
$\ker(\mathbf{X})$	Kernel of a matrix $\mathbf{X}$
$\mathrm{rank}(\mathbf{X})$	Rank of a matrix $\mathbf{X}$
$\mathrm{nullity}(\mathbf{X})$	Nullity of a matrix $\mathbf{X}$
$\mathrm{E}\{\mathsf{x}\}$	Expected value of x
$\Pr\{\mathsf{x} = x\}$	Probability that the random variable x is equal to x
$\triangleq$	Definition operator
$\gtrsim$	Approximately greater than
$\lesssim$	Approximately less than

Glossary

3GPP	3rd Generation Partnership Project
APP	Application layer
ARQ	Automatic repeat request
BEC	Binary erasure channel
BP	Belief propagation
DVB	Digital video broadcasting
EEP	Equal erasure protection
EW	Expanding window
FEC	Forward error correction
GE	Gaussian elimination
GNU	GNU's Not Unix!
GNU R	Programming language under the ↗**GNU** General Public License
HARQ	Hybrid ↗**ARQ**
IETF	Internet Engineering Task Force
IP	Internet protocol
IPDC	↗**IP** datacast
IPTV	↗**IP** television
ISO	International Organization for Standardization
LDGM	Low-density generator matrix
LDPC	Low-density parity-check
LT	Luby transform
MAP	Maximum a posteriori
MBMS	Multimedia broadcast/multicast services
MDS	Maximum distance separable
ML	Maximum likelihood
OSI	Open Systems Interconnection
PHY	Physical layer
Raptor	Rapid tornado
RFC	Request for comments
RLF	Random linear fountain

RS	Reed-Solomon
SEC	Symbol erasure channel
SPC	Single parity-check
LDPC	Low-density parity-check
LDRLF	Low-density random linear fountain
TCP	Transport control protocol
TTL	Time to live
UDP	User Datagram Protocol
UEP	Unequal error protection
VoIP	Voice over ↗**IP**

Chapter 1

Introduction

The Internet has become ubiquitous over the last two decades. The wide availability of high-speed internet connections have fostered the development of cheap or cost-free internet-based alternatives or extensions to traditional communications and media technologies like the telephone, radio or television. At the same time, traditional analogue technologies have been replaced by digital ones. Aside from the obvious economical reasons, the better digital means to guarantee a well-defined quality level, the better accessibility, a higher convenience and – not to forget about – the possibility to employ cryptographic measures against adversaries, have been further driving forces for the still ongoing digitalisation.

The deployment of digital technologies and their interlacement with the Internet have led to a predominance of packet-based information delivery. Such packet-switched communication networks enable the distribution of heterogeneous types of data to possibly many user devices with widely varying capabilities via the same infrastructure and support a variety of new services and applications with different fidelity and delay requirements for data delivery. As the data traffic generated by both stationary as well as mobile devices is steadily increasing, the limited available resources such as spectral bandwidth, data rate or admissible transmission power have to be utilised with increasing efficiency. Moreover, this multitude of communication scenarios is supposed to function reliably under extremely diverse and time-variant channel conditions.

1.1 From Algebraic to Probabilistic Channel Coding

The field of channel coding which is dedicated to answer the essential question of how to reliably and efficiently transmit information over a noisy channel has been sparked by Shannon with his ground-breaking paper “A Mathematical Theory of Communication” [Sha48]. Besides the derivation of the fundamental limit on the transmission rate over a noisy channel, i.e. the channel capacity, Shannon also proved the existence of digital channel codes that allow communication with an arbitrarily small error probability at any rate that does not exceed channel capacity. In fact he showed that almost any randomly chosen code achieves capacity as the blocklength, i.e. codeword length, goes to infinity. But since his proof was non-constructive in that a typical random code of large blocklength is prohibitively complex for practical implementation, he started a still ongoing quest for practical capacity-achieving channel codes.

In the first decades the developed codes were mostly of algebraic nature such as Hamming codes [Ham50], convolutional codes [Eli55] or Reed-Solomon (RS) codes [Bus52, RS60], but with the astounding invention of Turbo codes by Berrou et al. [BGT93, BG96] the gap to capacity could be dramatically diminished. Turbo codes, comprising a (pseudo-)random interleaver as an integral component, have drawn massive attention to probabilistic coding schemes and hence fuelled the development of modern coding theory.

Inasmuch as it is preposterous to common sense to put all one's eggs in one basket, it is unwise in coding to solely rely on individual extremely elaborated and complex codes. Instead, Turbo coding taught the lesson that simple constituent codes exchanging information in a well-thought-out randomised way achieve an error-correcting performance that individual codes are only able to reach at the price of a much higher complexity. Another important class of codes are low-density parity-check (LDPC) codes [Gal63]. These take to the extremes the principle of establishing a high global description complexity, which is required for good codes, by a multitude of quasi-randomly and iteratively interacting elementary constituent codes, i.e. repetition and single parity-check codes. Gallager has invented LDPC codes already in 1963 in his Ph.D. thesis. Except for a few further valuable contributions from Zyablov and Pinsker [ZP74], LDPC codes were largely forgotten, despite being way ahead of their time, since for those days the code complexity was far too high. In the wake of Turbo codes, however, they have been rediscovered independently by Mackay and Neal [MN95, MN96], Wiberg et al. [WLK95b, WLK95a] as well as Sipser and Spielman [SS96, Spi96]. Together with the so far mostly unnoticed contributions of Tanner [Tan81] that built the basis, the field "codes on graphs" arose as a conceptual unification of seemingly totally different types of codes, allowing numerous valuable results from random graph theory to find their way into coding theory.

Not noticing the importance of certain findings apparently has a long tradition in coding theory. Alongside with Gallager's LDPC codes and Tanner's bipartite graphs, the binary erasure channel (BEC) that was introduced by Elias in 1954 as a toy model of a communication channel was long seen as what it was intended to be, namely a toy model. Only very late it has found its matching part in the real communications world - the Internet or computer networks in general. A bit transmitted over a BEC is either correctly received or the bit is erased with a certain probability ϵ, but the bit is never in error. Similarly, a packet transmitted over the Internet is usually either received correctly or it is considered erased if it is lost or delayed due to congestions in the network or if it contains uncorrectable bit errors. Moreover, the BEC has proved invaluable in information theory. Properties and statements that hold on the BEC and which can be analysed, often in closed form, hold similarly on or give insights to scenarios with other well-known channels or channel models. The BEC and the binary symmetric channel (BSC) represent the two extremes of information combining [LHHH05, LH06].

1.2 Application Layer Forward Error Correction

Packet-based data transmission over communication networks is organised according to the Open Systems Interconnection (OSI) reference model [OSI94], a layered model of network architecture, standardised by the International Organization for Standardization (ISO). The model consists of seven logical layers, which provide protocols that are required to establish, maintain and terminate a communication session between different parties. Information can only be passed vertically between adjacent layers of the same instance or horizontally on the same layer between different instances. Lower layers provide services to upper layers and they interact only via well-defined protocols. In this thesis, only a few services are of further concern such as:

1. forward error correction (FEC) on the physical (PHY) layer
2. the cyclic redundancy check (CRC) on the logical link control (LLC) sublayer
3. and particularly FEC on the application (APP) layer.

The PHY layer provides a plethora of functions amongst which the most important for the current work is the establishment of a reliable bit-pipe for higher layers, the intermediate layers are required, e.g. for routing packets through the network, and the APP layer contains mostly source data processing components.

The PHY-FEC, however, is merely one of the first measures in the establishment of such a reliable bit-pipe. On the PHY layer error correcting codes operating on bit-level perform error correction mostly within a transmitted frame or packet and thereby combat noise or interferences that occur on the physical link. Usually, an additionally applied cyclic redundancy check (CRC) is used to test whether error correction has been successful or not. If decoding is successful, the decoded packet can be used in higher layers. If not, the packet is discarded and is considered erased. Recent, more intricate approaches as for instance in [Bre14] allow a higher permeability as well as iterative processing of reliability information between the layers and thereby increase the error robustness of the overall transmission. Though this approach may be extended up to the APP layer at the cost of a higher complexity, the current work, by focusing on FEC on the APP layer, considers the decision on the reliability of a packet as completed as it reaches the APP layer. Thus, the channel as it is seen from the APP layer's point of view is an erasure channel.

Despite the above-mentioned protection measures, there are several other reasons why a packet does not reach its destination, which cannot be compensated by a stronger PHY-FEC. One major reason is a congested network. Overwhelmed routers may discard packets because their buffer is full or it takes too long to redirect a packet. If the introduced delay exceeds the time to live (TTL) of a packet, it is discarded as well.

Most applications do not tolerate packet losses. In such cases, lost packets need to be resent in order to recover the original data. A well-known retransmission mechanism is automatic repeat request (ARQ), which uses a feedback channel to indicate that a certain packet is missing or could not be recovered and to initiate its retransmission. There exist also more sophisticated methods, such as the type II hybrid ARQ (HARQ) protocol, which send extra redundancy on the undecodable packet to a particular user, instead of retransmitting the original packet.

To protect the whole transmission against packet losses which mostly occur in the network and are hardly controllable at end user devices, nowadays usually the transport control protocol (TCP) is used. Essentially, TCP enables reliable point-to-point or unicast transmissions by requiring each packet of a message to be acknowledged by the receiver within a prescribed period of time. A missing or delayed acknowledgement means a packet is considered lost and entails countermeasures, i.e. the retransmission of unacknowledged packets.

With the increasing distribution of multimedia content or bulk data to a large number of end users, the delivery via multicast [3GP13] is becoming an attractive alternative to unicast transmission with a more efficient usage of the server and network resources. Instead of sending the same message to each user individually, all (subscribed) users are served at once, where routers determine the optimal paths to the destinations and create copies of the distributed packets if needed. However, severe problems like amplifying existing or even inducing new network congestions would arise if the transmission was based on a TCP-like protocol. First of all, the number of necessary packet acknowledgements scales linearly with the number of users which may become prohibitively large. Secondly, supplying each user individually with the respective missing packets is highly suboptimal. Since users often experience different and independent losses, the retransmitted packets are only useful to a specific user.

For applications with real-time character such as voice over Internet protocol (VoIP), an alternative transmission protocol, the user datagram protocol (UDP), is preferably used instead of TCP. In UDP resilience against packet loss has been sacrificed for low latency in that received packets are not acknowledged at the receiver and no retransmission is triggered if packets are missing. Merely a CRC is applied to prevent received packets containing bit errors to be further used. Though this protocol is in principle suited for broadcast or multicast delivery of, e.g. audio or video content, the lacking loss protection renders it ineligible for transmissions with a guaranteed quality level or for the transmission of bulk data, as the latter usually does not tolerate packet loss, unless some additional APP-FEC scheme is implemented. Such an APP-FEC scheme is discussed in the next section.

1.3 The Digital Fountain

For communication scenarios as described above, the “digital fountain approach” [BLMR98] constitutes a practical remedy for establishing an erasure-resilient data link on the APP without the need for feedback or with at most one acknowledging message per user upon the successful reception of the total transmission.

Ideally, this protocol should enable the recovery of a message that consists of k equally sized packets upon the reception of exactly k encoded packets of the same size in a transmission scenario as described above, irrespective of which k encoded packets have been received. Additionally, it should be possible to produce on the fly a potentially limitless number n of encoded packets from the k original packets. These properties include the notion of universality, i.e. the (near) optimal recovery of a message given an arbitrary erasure rate on the channel.

This protocol is called a “digital fountain” due to the analogy to a water fountain which is seen as an unlimited source of water allowing to fill the cups of many at the same time. Similar to filling one’s cup by collecting a sufficient number of nameless waterdrops, the original message should be decodable after collecting sufficiently many encoded packets, i.e. each encoded packet should equally and optimally contribute to the decodability. The corresponding erasure-resilient codes are called (digital) fountain codes or rateless codes. The latter name results from the property that their code rate $\rho = k/n$ is not determined a priori. Therefore, the terms “fountain code” and “rateless code” will be used synonymously.

A digital fountain is an idealised concept which in practice can only be approximated. Very early approaches for FEC schemes providing incremental redundancy [Man74, Dor83] are based on so-called maximum distance separable (MDS) [MS77] codes, i.e. codes that can recover a message that consists of k symbols (or packets) from any set of k out of n encoded symbols. Yet, MDS codes are not rateless and rateless codes, i.e. codes that allow to generate a potentially limitless number n of encoded symbols from a finite number of k uncoded symbols, are not MDS. Thus, in order to obtain practical fountain codes, the MDS condition is slightly alleviated [AL96] which allows to construct rateless erasure-resilient codes that are able to recover the message from any $k(1+\varepsilon)$ out of the n encoded symbols with high probability, where $\varepsilon \geq 0$ is a small relative overhead.

The first and still the most important class of practical rateless codes has been named Luby transform (LT) codes [Lub01, Lub02b, Lub02a] by their inventor

Michael Luby. LT codes are universal linear erasure-resilient irregular sparse-graph codes that are based on a particular random construction. They are designed to be iteratively decodable using the suboptimal belief propagation (BP) algorithm which is of low complexity. Online codes [May02] as well as Raptor codes [Sho04, Sho06, SLL06] represent further types of fountain codes that are constructed by combining a rateless code, e.g. an LT code, with one or more stages of high-rate precodes. The practical relevance of these codes has been recognised early on, so that today Raptor codes can be found in various communication standards like for instance IETF RFCs 5053 and 6330 [LSWS07, LSW$^+$11], the 3GPP MBMS standard [3GP13] for multimedia broadcasting and multicasting services, the DVB-IPDC standard [ETS09b] for IP datacast over digital video broadcasting networks or the DVB-IPTV standard [ETS09a] for TV services over IP networks. In general, precoding allows to use a weaker LT code than if it was used standalone while maintaining the erasure correction performance. Since the encoding and decoding complexity scales with the strength of an LT code, using a weaker code is a measure to achieve a lower complexity.

The main focus in the field of rateless codes has so far been on analysis and design of codes for large message sizes k or even under asymptotic assumptions ($k \to \infty$) as well as for using the low-complex but suboptimal BP decoding algorithm. Yet, if low delay applications are to be supported, only medium to short message sizes are suitable, but with a decreasing message size the erasure correction performance of the BP decoding algorithm degrades seriously. Instead, the optimal maximum likelihood (ML) decoding algorithm, which is usually considered too complex for large sizes, becomes affordable complexity-wise and at the same time almost imperative performance-wise for small to medium sizes. Thus, it will be the decoding method of choice in this thesis, as it significantly outperforms BP decoding in terms of the achieved erasure correction and entails an affordable computational effort in the small to medium size regime. Also new code designs are required that utilise the stronger capabilities of the ML decoder. Furthermore, since ML decoding is optimal in terms of erasure correction performance, the ML performance of a code can serve as an upper bound to any other decoding algorithm.

1.4 Designed Random Matrices over Finite Fields

If LT codes are used on an erasure channel, ML decoding corresponds to solving a consistent system of linear equations over a finite field, where the coefficients are given by the pruned LT code generator matrix. LT code generator matrices are based on a random construction consisting of several random processes of which perhaps the most important part is specified in terms of a particularly designed distribution of the matrix' non-zero coefficients, the so-called output node degree distribution or row weight distribution. The number of non-zero coefficients in each row, i.e. the row weight, is determined according to this distribution, defining thereby the expected erasure correction properties of the overall random matrix. In general, to determine the expected erasure correction performance of an LT code ensemble is equivalent to answering the fundamental mathematical question of whether a system of designed random linear equations over finite fields can be solved partially or completely.

1.5 Thesis Outline

The main objectives of this thesis are the analysis and the design of various types of rateless erasure-resilient code ensembles over finite fields with finite message sizes and under optimal erasure decoding. The thesis is structured as follows:

Chapter 2: Digital Fountain Codes

A general introduction to theoretical and practical aspects of digital fountain codes is provided with a particular focus on Luby transform (LT) codes as the most important realisation of a rateless erasure-resilient building block. Besides explaining the groundwork in the general non-binary domain, also systematic LT code ensembles are briefly addressed as well as Raptor codes, a concatenation of one or more high-rate precodes and a rateless component.

Chapter 3: Finite Length Analysis under Optimal Erasure Decoding

The core of this work, the finite length analysis and design of LT code ensembles under optimal erasure decoding, is presented in this chapter. A set of four bounds, consisting of upper and lower bounds on word and on symbol level on the residual erasure probability after optimal decoding, is derived in detail and is used to assess the performance of LT code ensembles or to design them efficiently without requiring extensive Monte Carlo simulations.

Thereafter, special LT code ensembles are designed, analysed and characterised. The class of sparse random LT code ensembles is devised and, under the given conditions, is identified to be quasi-optimal. The numerical evaluation of the proposed bounds and the comparison with Monte Carlo simulations provides further evidence for the theoretical claims. Finally, the computational complexity of some special LT code ensembles is assessed considering the variation of the field order, which leads to the new and intriguing conclusion that ensembles over moderately high field orders are superior to binary ones both in terms of erasure resilience as well as in terms of computational complexity.

Chapter 4: Conventionally Systematic LT Code Ensembles

Conventionally systematic LT code ensembles are characterised by an identity matrix prefix prepended to the common LT code generator matrix so that the input word is systematically contained in the output word. As briefly discussed in Section 2.2, in the literature such conventionally systematic ensembles are considered to be generally inferior to ensembles without the identity matrix prefix. It is shown that this conjecture does not hold in general and that some ensembles can benefit from this simple prefix by achieving a better erasure resilience at a lower computational complexity. Like in the previous chapter, a set of four bounds on the residual erasure probabilities is derived for conventionally systematic ensembles under optimal erasure decoding. Then, the theoretical results are evaluated for several examples and verified by Monte Carlo simulations.

Chapter 5: Precodes

Before dealing with another class of practically significant rateless code ensembles, i.e. Raptor code ensembles, suitable precodes as an essential ingredient need to be discussed first. So this chapter on precodes acts merely as a prologue to Chapter 6, in which several types of precodes, coarsely typified as deterministic and stochastic precodes, are analysed with respect to their erasure correction performance. Apart from having a high code rate, good precodes need to supply a strong protection particularly against low-weight erasure patterns. For some precodes, exact residual erasure probabilities on word level are either already known or are derived here. For others, like LDPC code ensembles, upper bounds are provided under optimal erasure decoding.

Chapter 6: Raptor Code Ensembles

A so-called erasure floor appears when few residual erasures, i.e. low-weight erasure patterns, are chiefly responsible for not fully recovering an information word. The limited performance of common LT code ensembles of practical complexity is characterised by a high erasure floor. Yet, by using the Raptor code construction, i.e. by employing a high rate precode that removes low-weight erasure patterns the erasure floor can be lowered efficiently. The bounds on the residual erasure probability for general LT code ensembles as provided in Chapter 3 can be easily used to assess the performance of LT code ensembles. This similarly applies to several precode types as discussed in Chapter 5. However, the derivation of corresponding bounds or equivalent performance measures for Raptor code ensembles is more involved.

Existing bounds from the literature turn out to be incorrect and are replaced by new quasi-bounds, a combination of the expressions of the used LT code ensemble and precode. Besides the derivation of the quasi-bounds, their validity range is discussed and the erasure correction performance of several example Raptor code ensembles is evaluated and compared with the quasi-bounds.

Chapter 7: Unequally Loss-Resilient LT Code Ensembles

Apart from such services as for instance plain data delivery where each bit is equally important, there exist other fields of applications like audio or video transmission which require an unequal protection of the unequally important parts of the data against erasures. Two general approaches from the literature for the construction of unequally loss-resilient LT code ensembles are discussed. First, the "weighting approach" is generalised to higher order Galois fields. In particular this approach is enhanced by a method called "biased sampling" which corrects a prior weakness and thus enables the achievability of any desired protection level for each importance class. For this "weighting approach with biased sampling" also the class specific set of four bounds on the residual erasure probability is derived. Secondly, the "expanding window approach" is generalised to higher order Galois fields and the class specific set of four bounds on the residual erasure probability is completed with bounds on word level. Finally, for unequally loss-resilient sparse random LT code ensembles constructed by the weighting method with biased sampling, a simplified heuristic design method is proposed.

Parts of the present thesis have been prepublished in the following papers which I have authored: [SSV11], [SLV11], [SL12], [SV12] and [SGV13]. These references are underlined throughout this thesis.

Chapter 2

Digital Fountain Codes

Digital fountain codes are a class of rateless erasure-resilient codes. They have first been characterised in [BLMR98], where also some application scenarios have been detailed, however, without an actual construction proposal. Initially, fountain codes have been stipulated for the binary erasure channel (BEC) (cf. Section 2.1.3 for the general channel model), for instance as an alternative to retransmission schemes such as automatic repeat request (ARQ). ARQ is usually required in packet-switched communication networks that are prone to packet losses in order to establish reliable communication.

A typical application of fountain codes is multicast, a system in which one transmitter broadcasts the same data to many subscribed users at the same time and in which the users experience different channel conditions and independent losses that are unknown to the transmitter. A particularly useful feature of fountain codes is that they do not require any knowledge of the erasure probability ϵ on the channel.

With fountain codes, the transmitter can generate an arbitrary and potentially infinite number n_T of encoded symbols $\mathbf{y}_\mathrm{T} = (y_1, y_2, \ldots, y_{n_\mathrm{T}})^\mathsf{T}$ from a finite number of k input symbols $\mathbf{x} = (x_1, x_2, \ldots, x_k)^\mathsf{T}$. Moreover, receivers should be able to decode the original k input symbols $\mathbf{x}$ from any $n_\mathrm{R} = k(1 + \varepsilon_\mathrm{R})$ received (i.e. unerased) encoded symbols with high probability if $\varepsilon_\mathrm{R} \geq 0$, where ε_R is the relative reception overhead[1].

For instance, to ensure a reliable multicast transmission as sketched in Figure 2.1, a conventional transmitter is usually tailored to the worst channel and inflicts a suboptimal use of channel resources upon receivers in better channel conditions. However, by using fountain codes, receivers with more favourable channel conditions do not need to continue to listen to the channel, but can just stop receiving additional encoded symbols as soon as they have successfully decoded the message.

So, although the transmission code rate $\rho_\mathrm{T} = \frac{k}{n_\mathrm{T}}$ can, in principle, reach zero, the reception code rate $\rho_\mathrm{R} = \frac{k}{n_\mathrm{R}} = \frac{1}{1+\varepsilon_\mathrm{R}}$ should be as close to one as possible to

[1] Note the typographical and semantic difference between the relative overhead ε and the erasure probability ϵ on the channel.

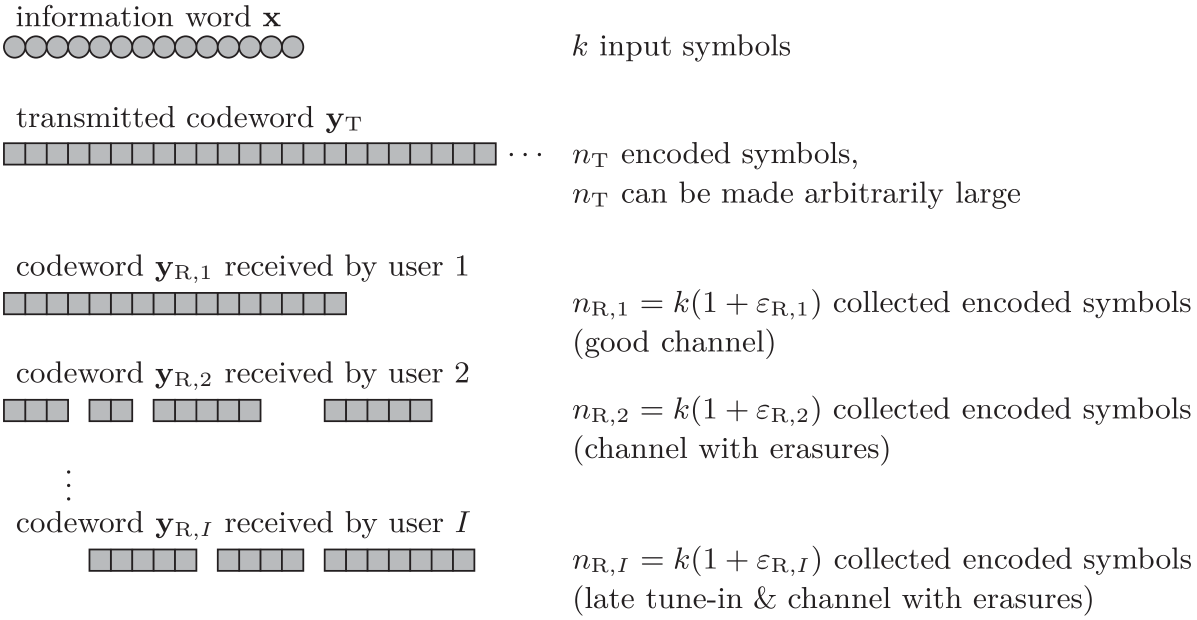

information word $\mathbf{x}$
k input symbols
transmitted codeword $\mathbf{y}_\mathrm{T}$
n_T encoded symbols,
n_T can be made arbitrarily large
codeword $\mathbf{y}_{\mathrm{R},1}$ received by user 1
$n_{\mathrm{R},1} = k(1+\varepsilon_{\mathrm{R},1})$ collected encoded symbols (good channel)
codeword $\mathbf{y}_{\mathrm{R},2}$ received by user 2
$n_{\mathrm{R},2} = k(1+\varepsilon_{\mathrm{R},2})$ collected encoded symbols (channel with erasures)
codeword $\mathbf{y}_{\mathrm{R},I}$ received by user I
$n_{\mathrm{R},I} = k(1+\varepsilon_{\mathrm{R},I})$ collected encoded symbols (late tune-in & channel with erasures)

Figure 2.1: Multicast scenario with fountain codes: due to the near-MDS property, each receiver is able to successfully decode the information word with high probability at the reception of slightly more than k encoded symbols. If the transmission time is sufficiently long, users can even tune in at different times.

approach capacity. In this receiver centric view, which is common in the fountain coding setup, a wasteful use of reception code rate ρ_R is penalised, not the use of the channel by the transmitter [STV07]. Nevertheless, in practice the use of channel resources is relevant and costly. So depending on the scenario, suitable transmission stopping criteria need to be applied, since it is not always feasible to continue the transmission until the user with the worst channel is satisfied. In a multicast or broadcast transmission this might lead to the situation that the transmission is stopped although a small fraction of users which are in very unfavourable channel conditions is not then able to decode the transmitted message. Yet although these considerations are highly relevant and should therefore be taken care of in practice, they are beyond the scope of this thesis.

In order to clearly differentiate between transmitter or receiver related quantities in the following, an index “T” or “R” is used if required. On the other hand, quantities that are stated in a general manner or which are equally related to both transmitter and receiver are written without an index.

2.1 Luby Transform (LT) Codes

Luby transform (LT) codes [Lub01, Lub02b, Lub02a] constitute the first and still most important type of practical and efficiently decodable linear rateless codes. They are low-density generator matrix (LDGM) codes without a fixed code rate.

LDGM codes are known as the dual of LDPC codes. LDGM codes usually have a rather limited performance. The limitation appears in terms of a high error floor or more precisely a high erasure floor, since by construction they contain some codewords of very low weight, i.e. with only few non-zero entries. An erasure floor is characterised by a very slow decrease of the residual erasure probability after decoding as additional encoded symbols are received. Nevertheless, the performance can be significantly improved if a precode is used to lower the erasure floor. Therefore, LDGM codes and LT codes as their rateless relatives are not intended to be used stand-alone but only in combination with a precode as will be discussed in Section 2.3 and Chapter 6.

2.1.1 LT Codes over Higher Order Galois Fields

Though the original LT codes were designed to encode binary symbols (i.e. bits), their recent generalisation to higher order Galois fields $\mathbb{F}_q$, where $q = 2^\mu$ and $\mu > 1$ [LPC10, QUA10, LSW+11, SLV11], has proven beneficial. The non-binary codes not only achieve substantial gains over their binary counterparts, but it can also be shown, that increasing the used Galois field to a moderate size can actually reduce the required decoding complexity [SLV11] significantly. Hence, the following description will be given for the general non-binary case. For a comprehensive introduction to finite fields, the reader is referred to the literature, e.g. [LNC97, Hub98].

An LT encoder is a linear map $\mathbb{F}_q^k \to \mathbb{F}_q^{n_\mathrm{T}}$ and is represented by an $n_\mathrm{T} \times k$ generator matrix $\mathbf{G}_\mathrm{T}$ over $\mathbb{F}_q$, i.e. $\mathbf{G}_\mathrm{T} \in \mathbb{F}_q^{n_\mathrm{T} \times k}$, where $n_\mathrm{T} \to \infty$. The k input symbols $\mathbf{x} \in \mathbb{F}_q^k$ are mapped to n_T output or encoded symbols $\mathbf{y}_\mathrm{T} \in \mathbb{F}_q^{n_\mathrm{T}}$ by

$$\mathbf{y}_\mathrm{T} = \mathbf{G}_\mathrm{T}\mathbf{x}. \tag{2.1}$$

Although an LT encoder is a linear map to an infinite-dimensional vector space, i.e. with $n_\mathrm{T} \to \infty$, in practice the number n_T of created encoded symbols is kept finite according to some predefined stopping criteria.

The LT code generator matrix $\mathbf{G}_\mathrm{T}$ defines the edges of a bipartite graph that connect the input nodes, which represent the input symbols, to the output nodes, that represent the encoded symbols. Since erasure information is usually given on packet level, the graph with the smallest input size k associates the symbols x_i and y_j, where $i \in \{1, 2, \ldots, k\}$ and $j \in \{1, 2, \ldots, n_\mathrm{T}\}$, directly with a whole *packet* of data as depicted in Figure 2.2. A packet consists of l $\mathbb{F}_q$-elements, where an $\mathbb{F}_q$-element is represented by a small circle or square in Figure 2.2. However, the graph is independent of the symbol size l and so this number l has no influence on the erasure correction performance of a fixed but arbitrary code [Lub02a]. Therefore, $l = 1$ is assumed throughout this thesis.

Naturally, $\mathbb{F}_q$-elements have an equivalent binary representation which requires $\mu = \mathrm{ld}(q)$ bits per element. In order to ensure a fair comparison between codes

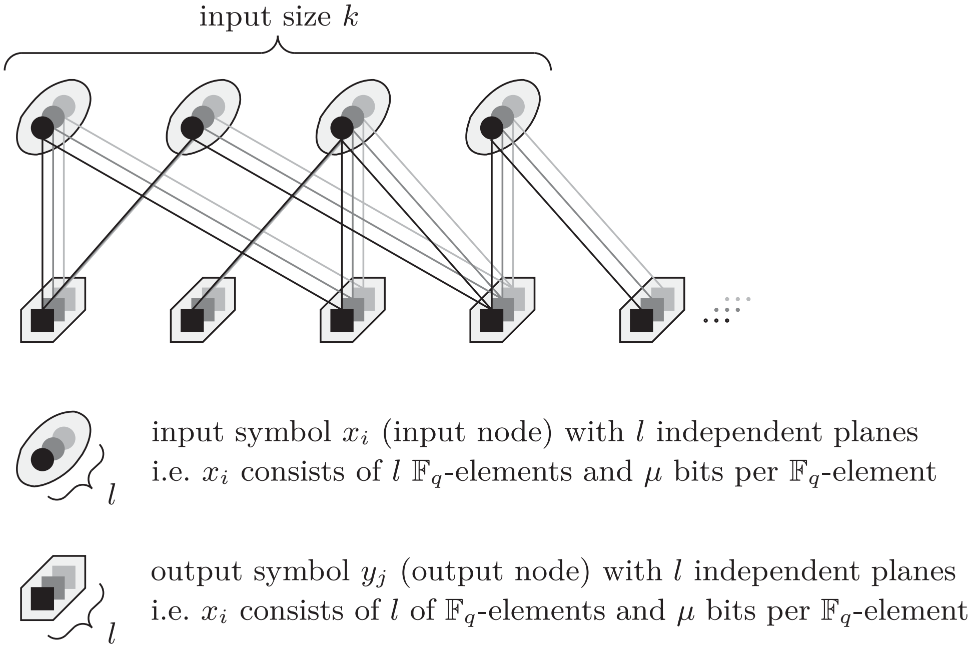

input size k
input symbol x_i (input node) with l independent planes
i.e. x_i consists of l $\mathbb{F}_q$-elements and μ bits per $\mathbb{F}_q$-element
l
output symbol y_j (output node) with l independent planes
i.e. x_i consists of l of $\mathbb{F}_q$-elements and μ bits per $\mathbb{F}_q$-element
l

Figure 2.2: General LT code graph with l independent planes of $\mathbb{F}_q$-elements and μ bits per $\mathbb{F}_q$-element, where $\mu = \mathrm{ld}(q)$.

over Galois fields of different orders, the number k_{B} of input *bits* is kept constant hereafter. Consequently, the number of input *symbols* is $k = \lceil \frac{k_{\mathrm{B}}}{\mathrm{ld}(q)} \rceil = \lceil \frac{k_{\mathrm{B}}}{\mu} \rceil$, i.e. the input size of a code over $\mathbb{F}_q$ with $q = 2^{\mu}$ is k. This is illustrated in Figure 2.3.

In contrast to traditional block codes, the matrix $\mathbf{G}_{\mathrm{T}}$ is generated online and can differ for each data block. The decoder is assumed to be aware of the connections between each output and input symbol, i.e. the current matrix $\mathbf{G}_{\mathrm{T}}$ is known. This can be achieved, e.g. by synchronising identical pseudo-random processes that produce $\mathbf{G}_{\mathrm{T}}$.

2.1.2 LT Code Construction and the Row Weight Distribution

The erasure correction performance of an LT code is mainly determined by a probability mass function (pmf) on the row weight, which is sometimes also called output node degree distribution (from node perspective). This row weight distribution $\Omega_0, \Omega_1, \dots, \Omega_k$ is defined on the finite set of numbers $\{0, 1, \dots, k\}$, where a row has weight d with probability Ω_d, i.e. the corresponding output node has d edges that connect it to d distinct input nodes, chosen uniformly at random without replacement from the set of k input nodes. However, in practice, the coefficient Ω_0 should be set to zero, since unconnected output nodes are useless. Often the row weight (or degree) distribution is not defined on the complete set of numbers $\{0, 1, \dots, k\}$ but on a smaller row weight (or degree) sample space $\mathcal{D}$ that contains only those numbers d for which $\Omega_d \neq 0$. Typically, the row weight distribution is

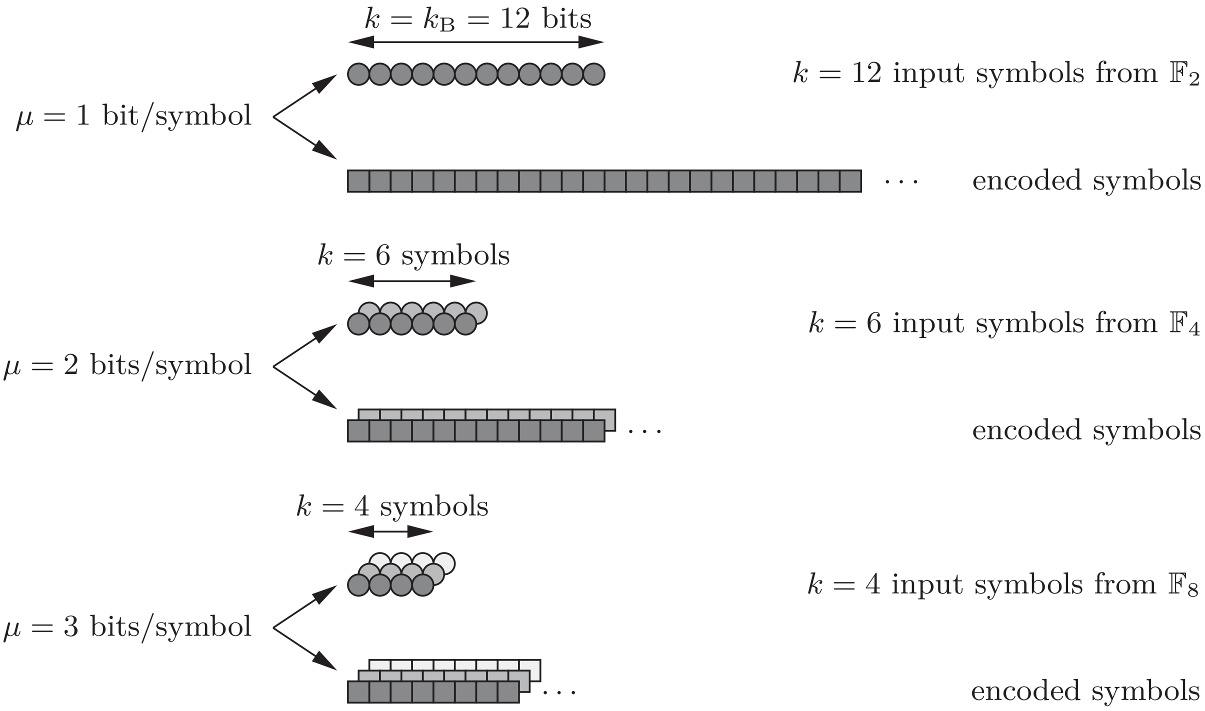

$k = k_\mathrm{B} = 12$ bits
$\mu = 1$ bit/symbol
$k = 12$ input symbols from $\mathbb{F}_2$
encoded symbols
$k = 6$ symbols
$\mu = 2$ bits/symbol
$k = 6$ input symbols from $\mathbb{F}_4$
encoded symbols
$k = 4$ symbols
$\mu = 3$ bits/symbol
$k = 4$ input symbols from $\mathbb{F}_8$
encoded symbols

Figure 2.3: To ensure a fair comparison of rateless codes over Galois fields of different orders, the number k_B of bits in the information word is kept constant, while the input size, i.e. the number of input symbols, is $k = \lceil \frac{k_\mathrm{B}}{\mathrm{ld}(q)} \rceil = \lceil \frac{k_\mathrm{B}}{\mu} \rceil$.

given in terms of its generator polynomial

$$\Omega(\xi) = \sum_{d=0}^{k} \Omega_d \, \xi^d = \sum_{d \in \mathcal{D}} \Omega_d \, \xi^d. \tag{2.2}$$

In the generator matrix $\mathbf{G}_\mathrm{T}$ the d non-zero entries in a row correspond to the values assigned to the d edges between an output node and d input nodes. The value of an output node is determined by the weighted sum of the connected d input nodes, where the weights are given by the respective connecting edges. The non-zero entries of $\mathbf{G}_\mathrm{T}$ are chosen uniformly from the set of $q-1$ non-zero $\mathbb{F}_q$-elements. Note that in the following a different font is used when referring to random variables in order to underline the random nature of certain quantities. For instance, the random generator matrix is denoted $\boldsymbol{\mathsf{G}}_\mathrm{T}$ instead of $\mathbf{G}_\mathrm{T}$, the latter is just a particular realisation.

Two important quantities are the average row weight or the average output node degree

$$\bar{d} = \sum_{d \in \mathcal{D}} d\Omega_d \tag{2.3}$$

as well as the expected density

$$\Delta = \bar{d}/k \tag{2.4}$$

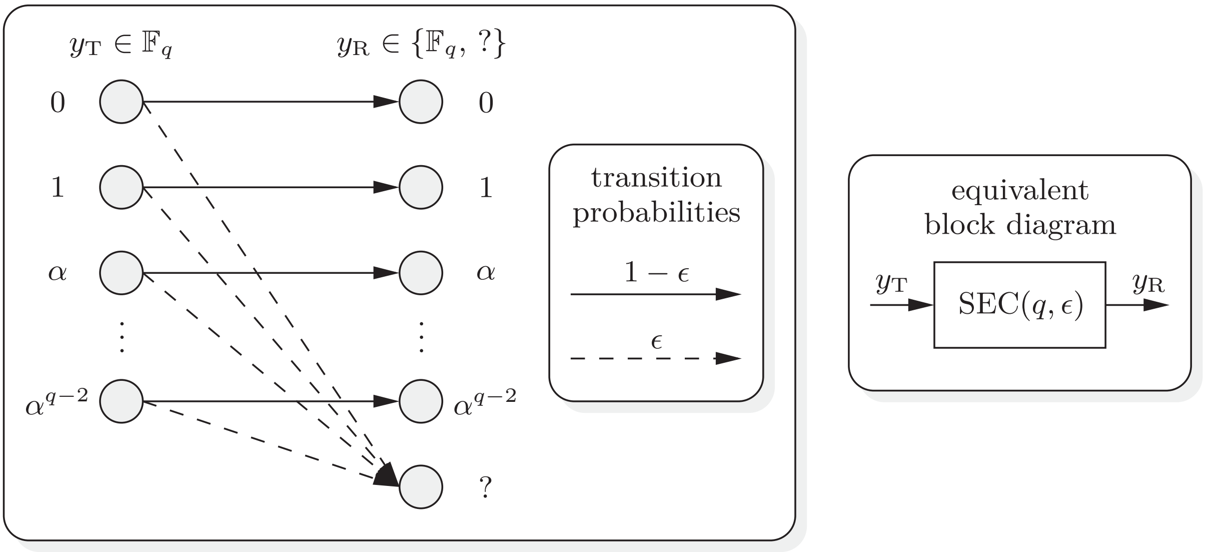


Figure 2.4: q-ary symbol erasure channel ($\mathrm{SEC}(q,\epsilon)$) with symbol erasure probability ϵ.

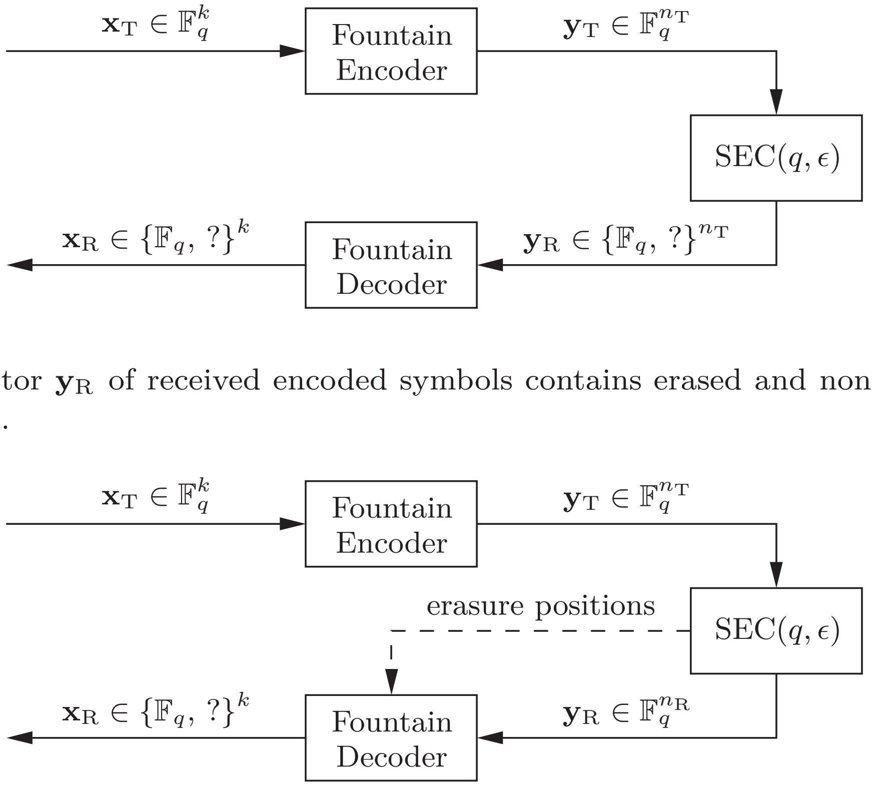
of a random matrix with $0 \leq \Delta \leq 1$, which corresponds to the expected relative amount of non-zero elements. In the following, the expected density will only be referred to as density. These two closely related quantities are good measures for the encoding and the decoding complexity.

2.1.3 The Symbol Erasure Channel

The encoder produces n_{T} output symbols that are transmitted over a symbol erasure channel (SEC) that randomly erases some of these transmitted encoded symbols. The SEC, as depicted in Figure 2.4, is a straightforward generalisation of the BEC, which is required if codes on higher order Galois fields are used. Transmitted symbols are received correctly with probability $1-\epsilon$ or are erased with probability ϵ. An erased symbol is marked by "?". The SEC is supposed to abstract and incorporate all sources of packet (or symbol) losses which occur on lower protocol layers.

At the receiver, $n_{\mathrm{R}} \leq n_{\mathrm{T}}$ encoded symbols are collected from which the decoder tries to reproduce the original k input symbols. In Figure 2.5 two equivalent simplified transmission chains, consisting of a fountain encoder, an SEC and a fountain decoder, are depicted together with the symbol vectors and their dimensions. The encoded symbol vector $\mathbf{y}_{\mathrm{R}}$ on the receiver side in Figure 2.5(a) may contain erasures, while in Figure 2.5(b) erased encoded symbols are pruned from $\mathbf{y}_{\mathrm{R}}$ such that it is shorter than $\mathbf{y}_{\mathrm{T}}$ by the number of occurred erasures. The pruned positions are assumed to be known to the receiver. Only the latter transmission model will be used hereafter.

Having collected $n_{\mathrm{R}} \leq n_{\mathrm{T}}$ output symbols, the decoder uses the n_{R} rows of $\mathbf{G}_{\mathrm{T}}$ that are associated with the received, i.e. the collected, non-erased encoded symbols to


$\mathbf{x}_\mathrm{T} \in \mathbb{F}_q^k$
Fountain Encoder
$\mathbf{y}_\mathrm{T} \in \mathbb{F}_q^{n_\mathrm{T}}$
SEC(q, ϵ)
$\mathbf{x}_\mathrm{R} \in \{\mathbb{F}_q, ?\}^k$
Fountain Decoder
$\mathbf{y}_\mathrm{R} \in \{\mathbb{F}_q, ?\}^{n_\mathrm{T}}$
$\mathbf{x}_\mathrm{T} \in \mathbb{F}_q^k$
Fountain Encoder
$\mathbf{y}_\mathrm{T} \in \mathbb{F}_q^{n_\mathrm{T}}$
erasure positions
SEC(q, ϵ)
$\mathbf{x}_\mathrm{R} \in \{\mathbb{F}_q, ?\}^k$
Fountain Decoder
$\mathbf{y}_\mathrm{R} \in \mathbb{F}_q^{n_\mathrm{R}}$

(a) The vector $\mathbf{y}_\mathrm{R}$ of received encoded symbols contains erased and non-erased symbols.

(b) Erased symbols are pruned from the vector $\mathbf{y}_\mathrm{R}$ of received encoded symbols, i.e. it contains only the n_R non-erased encoded symbols. It is assumed that the fountain decoder is aware of the pruning positions. This transmission model is used throughout this thesis.

Figure 2.5: Symbol vectors and their dimensions in the transmission chain.

make up a new matrix $\mathbf{G}_\mathrm{R}$ on which decoding is performed. Since $\mathbf{G}_\mathrm{R}$ consists of a set of n_R rows sampled at random from the original matrix $\mathbf{G}_\mathrm{T}$ according to the erasures that occur on the SEC, G_R follows the same row weight distribution as G_T, i.e. if $\mathsf{G}_\mathrm{T} \sim \Omega(\xi)$, also $\mathsf{G}_\mathrm{R} \sim \Omega(\xi)$. Often, the number of transmitted or received encoded symbols will be expressed in terms of the relative overhead ε_T or ε_R, or in terms of the inverse code rate γ_T or γ_R, i.e.

$$n_\mathrm{T} = k(1 + \varepsilon_\mathrm{T}) = k\gamma_\mathrm{T} \qquad \text{or} \qquad n_\mathrm{R} = k(1 + \varepsilon_\mathrm{R}) = k\gamma_\mathrm{R}. \tag{2.5}$$

2.1.4 Ensembles

Given a row weight distribution $\Omega(\xi)$ on $\mathcal{D}$ (or equivalent construction constraints), as well as the dimension of the domain $\mathbb{F}_q^k$ and of the codomain $\mathbb{F}_q^n$, the sample space $\mathcal{G} \subseteq \mathbb{F}_q^{n\times k}$ of LT code generator matrices $\mathbf{G}$ contains

$$|\mathcal{G}| = \left(\sum_{d\in\mathcal{D}} \binom{k}{d} (q-1)^d \right)^n \tag{2.6}$$

different matrices, which holds for both the transmitter and the receiver perspective. For a row of weight d there are $\binom{k}{d}$ possibilities to distribute d non-zero entries to k positions and each non-zero entry can be chosen from the set of $q-1$ non-zero $\mathbb{F}_q$-elements. Moreover, there are n independent rows in $\mathbf{G}$.

Note that in general the generator matrices $\mathbf{G} \in \mathcal{G} \subseteq \mathbb{F}_q^{n\times k}$ do not have equal probabilities of occurrence and thus, an additional quantity is required, namely a probability distribution $\Pr\{\boldsymbol{\mathsf{G}} = \mathbf{G}\}$ on this set $\mathcal{G}$. This pair defines an (LT code) ensemble $\mathfrak{L} = (\Pr\{\boldsymbol{\mathsf{G}} = \mathbf{G}\}, \mathcal{G})$. An equivalent, yet less specific, definition is $\mathfrak{L} = \left(\Pr\{\boldsymbol{\mathsf{G}} = \mathbf{G}\}, \mathbb{F}_q^{n\times k}\right)$. However, since such a probability distribution on $\mathcal{G}$ is usually not given explicitly but implicitly by the row weight distribution $\Omega(\xi)$ as indicated by the notation $\boldsymbol{\mathsf{G}} \sim \Omega(\xi)$, an LT code ensemble is preferably given by $\mathfrak{L} = (\boldsymbol{\mathsf{G}} \sim \Omega(\xi), \mathcal{G})$ or $\mathfrak{L} = \left(\boldsymbol{\mathsf{G}} \sim \Omega(\xi), \mathbb{F}_q^{n\times k}\right)$. If the dimensions k and n or the sample space $\mathcal{G}$ are clear from the context, it is sufficient to refer to an ensemble $\mathfrak{L}$ just by its row weight distribution $\Omega(\xi)$.

Like for their fixed-rate relatives, i.e. LDPC codes, it is common in the field of rateless codes to investigate and design good ensembles before designing good concrete codes. The reason for this is the so-called concentration effect, which means that the performance of a code $\mathbf{G}$, chosen from an ensemble with probability $\Pr\{\mathbf{G}\}$, concentrates around the average ensemble performance with high probability. Since LT codes are supposed to generate any desired and possibly large number of encoded symbols, only the first part of an encoding matrix can in practice be equipped with an intricate structure and special properties. The remaining rows may then as well be constructed randomly according to some properly designed distribution $\Omega(\xi)$. Additionally, for an increasing erasure probability ϵ on the channel, the structure in the first part of the matrix becomes more and more invisible to the receiver and its performance quickly converges to the ensemble average. And therefore, the focus of this thesis is mainly on unstructured ensembles.

2.1.5 Decoding Algorithms

For erasure channels, there exist essentially two different decoding algorithms, namely the greedy, yet suboptimal belief propagation (BP) decoding algorithm and the optimal, yet computationally more complex maximum likelihood (ML) decoding algorithm. Both will be briefly reviewed in the following.

Belief Propagation Decoding

The belief propagation (BP) decoding algorithm is also known as greedy decoding or, as is the case here, in conjunction with erasure channels it is known as peeling decoding. It is best explained by using the decoding graph, i.e. the bipartite graph that represents the relationship between the input symbols and the received output symbols. A detailed example of BP decoding of an LT code over $\mathbb{F}_4$ can be found in Figure 2.6 illustrating the individual steps of the BP algorithm listed below.

1. Find an output node of degree one. If none can be found, decoding fails and eventually more output nodes have to be collected to restart decoding.

2. Propagate the selected output node's value to the connected input node by dividing the output node's value by the weight of the connected edge.

3. Remove the used output node and its edge from the decoding graph.

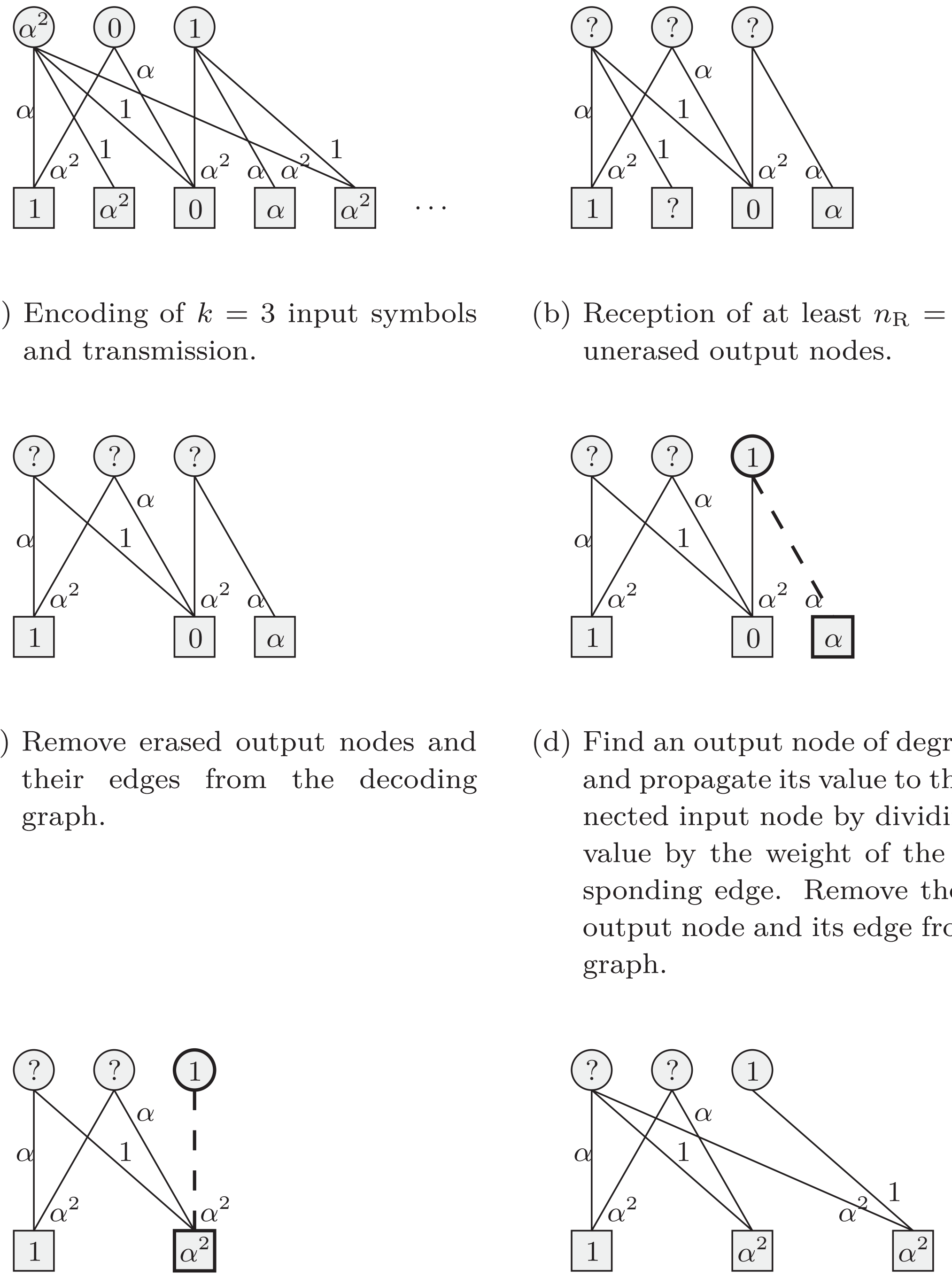
4. Propagate the recovered input node's value to all connected output nodes. This is done by adding to each output node's value the input node's value multiplied by the weight of the respective connecting edge.

5. Remove all used edges from the decoding graph. If all input nodes have been recovered, decoding ends successfully. If there still exist undecoded input nodes, continue with step 1.

Since each edge in the decoding graph is used at most once, the complexity of recovering an information word by BP decoding is proportional to the average row weight $\bar{d}$ and the number of input nodes k, i.e. $\mathcal{O}(\bar{d}k)$. This low complexity makes BP decoding very interesting for practical applications.

BP decoding for erasure channels corresponds to solving a consistent system of linear equations solely by means of back substitution. Thus, it is clear that if the corresponding matrix $\mathbf{G}_\mathrm{R}$ can be transformed into an upper triangular matrix just by column and row permutations, BP decoding succeeds and delivers the optimal ML solution. If on the other hand $\mathbf{G}_\mathrm{R}$ can be upper triangulated only partially by column and row permutations, BP decoding can recover merely those input symbols that are associated with the upper triangular part of $\mathbf{G}_\mathrm{R}$ and which do not depend on only optimally, i.e. ML decodable symbols. The other input symbols can, if at all, be solved solely by optimal decoding.

For properly designed codes and especially for large input sizes, the suboptimal BP algorithm works remarkably well. However, for small to medium input sizes the variance of the random process that creates degree one output nodes during decoding is too high. So decoding very often fails due to the lack of degree one output nodes and this leads to a too high overhead of additionally received symbols that is required for successful decoding, and thus in this domain of input sizes, optimal decoding is left as the only sensible choice.


α^2 0 1
α 1 1 α^2 α α^2 α α^2 1
1 α^2 0 α α^2 . . .
? ? ?
α 1 1 α^2 α α^2 α
1 ? 0 α
? ? ?
α 1 α^2 α α^2 α
1 0 α
? ? 1
α 1 α^2 α α^2 α
1 0 α
? ? 1
α 1 α^2 α α^2
1 α^2
? ? 1
α 1 α^2 α α^2 1
1 α^2 α^2

(a) Encoding of $k = 3$ input symbols and transmission.

(b) Reception of at least $n_{\mathrm{R}} = k = 3$ unerased output nodes.

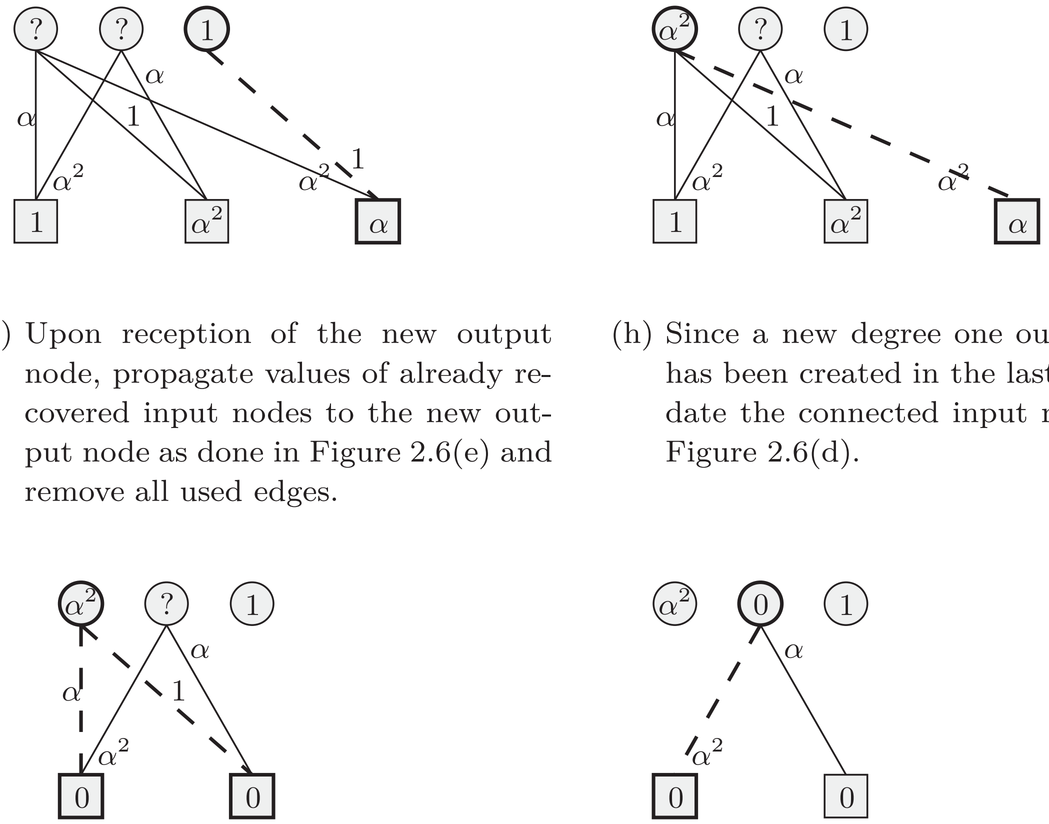
(c) Remove erased output nodes and their edges from the decoding graph.

(d) Find an output node of degree one and propagate its value to the connected input node by dividing the value by the weight of the corresponding edge. Remove the used output node and its edge from the graph.

(e) Propagate the value of the recovered input node to all connected output nodes by multiplying its value by the weights of the respective edges and add these values to the values of the respective output nodes. Remove the used edges.

(f) Since no degree one output node has been created in the decoding process, additional output nodes have to be collected.

Figure 2.6: Exemplary belief propagation decoding of an LT code over $\mathbb{F}_4$. The illustration is continued in Figures 2.6(g) – 2.6(j) on the next page. Ancillary addition and multiplication tables for $\mathbb{F}_4$ are provided in Table 2.1.



(g) Upon reception of the new output node, propagate values of already recovered input nodes to the new output node as done in Figure 2.6(e) and remove all used edges.

(h) Since a new degree one output node has been created in the last step, update the connected input node as in Figure 2.6(d).

(i) Perform an output node update as in Figure 2.6(e) and remove all used edges.

(j) Either of the two degree one output nodes can be used to perform the final input node update as described in Figure 2.6(d).

Figure 2.6: Exemplary belief propagation decoding of an LT code over $\mathbb{F}_4$. Continuation from previous page.

Table 2.1: Addition and multiplication tables for $\mathbb{F}_4$, which is defined by the primitive polynomial $a(\xi) = 1 + \xi + \xi^2$ in $\mathbb{F}_2$. The primitive element is denoted α.

$+$	0	1	α	α^2
0	0	1	α	α^2
1	1	0	α^2	α
α	α	α^2	0	1
α^2	α^2	α	1	0

$\cdot$	0	1	α	α^2
0	0	0	0	0
1	0	1	α	α^2
α	0	α	α^2	1
α^2	0	α^2	1	α

Maximum Likelihood Decoding

The maximum likelihood (ML) decoding algorithm for erasure channels coincides with the maximum a posteriori (MAP) decoding algorithm. It is the optimal decoding algorithm in terms of decoding failure probability. ML decoding of LT codes on erasure channels is equivalent to solving a system of n_{R} consistent linear equations in k unknowns over a finite field $\mathbb{F}_q$. If the respective coefficient matrix $\mathbf{G}_{\mathrm{R}}$ has full column rank, i.e. $\mathrm{rank}(\mathbf{G}_{\mathrm{R}}) = k$, an arbitrary information word can be uniquely determined. If $\mathrm{rank}(\mathbf{G}_{\mathrm{R}}) < k$, the solution of $\mathbf{G}_{\mathrm{R}}\mathbf{x} = \mathbf{y}_{\mathrm{R}}$ spans a $(k - \mathrm{rank}(\mathbf{G}_{\mathrm{R}}))$-dimensional vector space.

Such a system can be solved for instance by means of the well-known Gaussian elimination (GE) algorithm. Since GE has a relatively high computational complexity of $\mathcal{O}(k^3)$ per information word, ML decoding is practically only applicable to codes with short to medium input sizes. However, as codes for short to medium input sizes are highly relevant for low-delay applications and have not been well investigated so far, this thesis concentrates on the finite length analysis of LT codes and Raptor codes under ML decoding. Moreover, LT ensembles are proposed that have a near-optimal erasure correction performance under ML decoding, given certain design constraints.

Efficient Maximum Likelihood Decoding Algorithms

Besides GE there exist several algorithms that achieve the ML erasure correction performance but eventually exploit certain properties (cf. e.g. [Wie86, LO91, PS92, Cop93, RU01, Cop94, BM04, PNF04, SLK05, LMPC09, PLMC12]) such as the sparseness of the matrix $\mathbf{G}_{\mathrm{R}}$ to reduce the number of operations and thus the computational complexity. The main difference to GE is the scheduling according to which operations are carried out. Nonetheless, the complexity of these algorithms is still upper bounded by $\mathcal{O}(k^3)$.

A simple, yet effective method is the consecutive use of BP decoding followed by ML decoding by means of GE. If degree one output nodes exist, BP decoding can reduce the number of unknowns from k to $\tilde{k}$ with complexity $\mathcal{O}(\bar{d}(k - \tilde{k}))$ and the remaining $\tilde{k}$ unknowns can be solved by GE with complexity $\mathcal{O}(\tilde{k}^3)$. Although this concatenation of BP and ML decoding may not be the fastest algorithm, it is the method of choice in this thesis. Despite its simplicity it contains already the ingredient for the main part of the speed-up, i.e. reducing the original number of unknowns as far as possible by means of a low complexity algorithm and solve the rest by GE. Also it facilitates comparability with results from the literature. Moreover, it is only used to quantify *relative* decoding speeds of given LT code ensembles, which should vary only slightly when applying other efficient ML decoding algorithms.

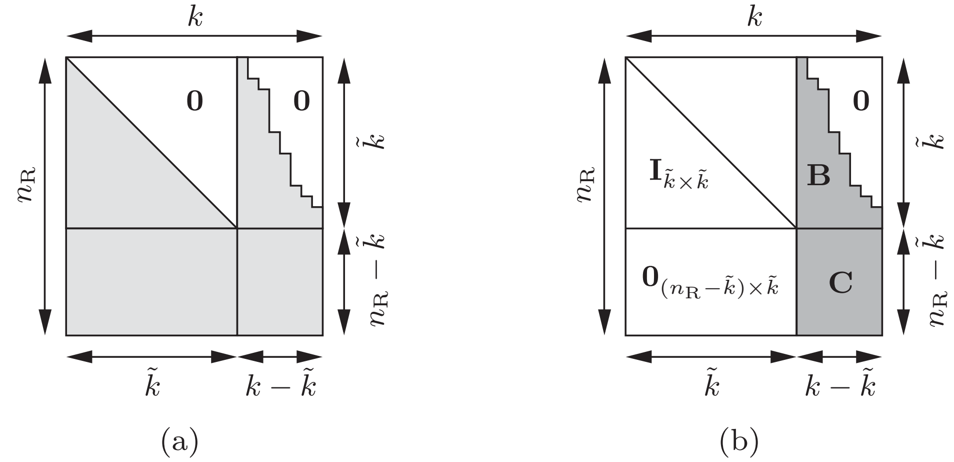


Figure 2.7: Decoding matrix $\mathbf{G}_\mathrm{R}$ during inactivation decoding (cf. [SL11]).

So far, in practical systems such as the 3GPP Multimedia Broadcast/Multicast Service (MBMS) [3GP13], where Qualcomm's Raptor10$^\mathrm{TM}$ [LSWS07] and RaptorQ$^\mathrm{TM}$ [QUA10,LSW$^+$11,SL11] codes are used, another efficient ML decoding algorithm named inactivation decoding [SLK05] is employed.

In contrast to the previously described simple method of first applying BP decoding and then GE, inactivation decoding consists of multiple steps, of which the first one is to rearrange the system of linear equations. This is accomplished by means of row and column permutations on the decoding matrix $\mathbf{G}_\mathrm{R}$ to transform it into a matrix that is partitioned like in Figure 2.7(a) and to maximise the size $\tilde{k}$ of the lower triangular submatrix. The original decoding matrix $\mathbf{G}_\mathrm{R}$ is supposed to have a low density. In the next step, row additions are performed in order to eliminate the entries on the left below the main diagonal of the decoding matrix. The result of this step is sketched in Figure 2.7(b). By row additions, the submatrices $\mathbf{B}$ and $\mathbf{C}$ usually become dense, which is indicated by a darker grey. The subsystem of linear equations defined by submatrix $\mathbf{C}$ is solved using GE. It can only be solved if $\mathbf{C}$ has full column rank. Then, the recovered unknowns can be used to solve the remaining unknowns using back substitution, i.e. BP decoding, which eliminates the entries in submatrix $\mathbf{B}$. A more detailed description of inactivation decoding can be found in [SLK05, SL11].

2.1.6 Special Row Weight Distributions

The row weight distribution $\Omega(\xi)$ is the only quantity that determines the performance of an LT code ensemble after having received a certain number n_R of encoded symbols and considering the number k of input symbols, the field size q as well as the decoding algorithm as given.

For BP decoding, there exist several row weight distributions, all aiming at optimising the size of the so-called ripple. During BP decoding, the ripple is the set of output nodes of current degree one. Since decoding fails as soon as the ripple runs empty, it is extremely important to ensure a positive ripple size over the whole decoding process. Overly large ripples on the other hand are wasteful and should be avoided as well.

The Soliton Distributions

The so-called ideal soliton distribution

$$\Omega_d = \begin{cases} \frac{1}{k} & \text{if } d = 1 \\ \frac{1}{d(d-1)} & \text{if } 2 \leq d \leq k, \end{cases} \tag{2.7}$$

proposed by Luby [Lub02a], is merely of theoretical relevance, as it has an expected ripple size of one. Due to the non-zero variance of the ripple size during BP decoding ripple sizes greater than one do occur and consequently, with high probability, also a ripple size of zero before decoding has finished, which immediately results in a decoding failure. Its average degree is known to be equal to the k^{th} harmonic number, i.e. $\bar{d} = \sum_{d=1}^{k} \frac{1}{d}$. If $k \longrightarrow \infty$, the average degree increases logarithmically and converges to $\ln k + 0.57721\ldots$, where $0.57721\ldots$ is the Euler–Mascheroni constant.

The robust soliton distribution, also proposed by Luby [Lub02a], is a more stable version of the ideal soliton distribution with a higher expected ripple size. This leads to a much lower decoding failure probability.

A Row Weight Distribution Optimised for BP Decoding

In [Sho06] Shokrollahi presented a semi-heuristic method to optimise row weight distributions for precoded LT codes, i.e. Raptor codes, for BP decoding. This method also comprises the reasoning behind the soliton distributions. The asymptotic design makes use of the so-called and-or-tree analysis [LMS98] which these days is better known by the name "density evolution", while the finite length design is a heuristic modification that tries to ensure that the decoding process continues until all but a certain fraction of LT code input symbols have been recovered with high probability. In this thesis, the degree distribution

$$\begin{aligned} \Omega(\xi) = {} & 0.007969\xi + 0.49357\xi^2 + 0.16622\xi^3 + 0.072646\xi^4 \\ & + 0.082558\xi^5 + 0.056058\xi^8 + 0.037229\xi^9 \\ & + 0.05559\xi^{19} + 0.025023\xi^{65} + 0.003135\xi^{66} \end{aligned} \tag{2.8}$$

from [Sho06], though designed for BP decoding and a quite large input size of $k = 65536$, will be frequently used as a reference, especially to illustrate differences to row weight distributions that are better suited for ML decoding or to allow for a better comparison of presented results with results from the literature. The average row weight of this ensemble amounts to $\bar{d} = 5.87$.

The Standard and the Sparse Random Ensemble

Properties of random matrices whether sparse or dense, on the reals as well as on finite fields are of particular importance for numerous disciplines such as coding theory, statistical physics and computer science, to mention just a few. In this thesis, only the properties of random matrices on finite fields will be investigated, as only these are relevant to the considered LT code setup.

The random matrices that are mostly examined in the literature are created by an *element-wise* independent random process, i.e. a Bernoulli process. The binary standard random ensemble [Mac05] is generated such that each entry in the matrix is chosen independently and uniformly at random from $\mathbb{F}_2$ or more generally in case of the standard random ensemble the entries are chosen from $\mathbb{F}_q$, i.e. each element from $\mathbb{F}_q$ occurs with equal probability $1/q$. Whereas for the sparse random ensemble a bias is introduced regarding the zero element, which is assigned a higher probability than each non-zero $\mathbb{F}_q$-element. Especially the standard ensemble, but partly also the sparse ensemble, plays a special role not only in the realm of LT codes, but also for instance in random linear network coding.

The standard random ensemble is sometimes also denoted conventional random linear fountain (RLF) ensemble. And the sparse random ensemble is also called low-density (or sparse) random linear fountain (LDRLF) ensemble. The respective terms will be used synonymously. However, in general, LT code generator matrices are created according to a *row-wise* independent random process which is specified by a given row weight distribution $\Omega(\xi)$. Therefore, to be compliant with the creation process of arbitrary LT codes, which additionally offers a stronger influence on the properties of a matrix, the random ensembles will also be constructed row-wise using the row weight distributions $\Omega(\xi)$ that result from the element-wise construction.

The row weight distribution of the standard random ensemble (or conventional RLF ensemble) amounts naturally to the binomial distribution

$$\begin{aligned}\Omega(\xi) &= \sum_{d=0}^{k} \binom{k}{d} \left(\frac{q-1}{q}\right)^{d} \left(1 - \frac{q-1}{q}\right)^{k-d} \xi^{d} \\ &= \frac{1}{q^k} \sum_{d=0}^{k} \binom{k}{d} (q-1)^{d} \xi^{d}. \qquad (2.9)\end{aligned}$$

Using the row weight distribution in (2.9) or the element-wise standard random construction is equivalent. The row weight distribution just defines the probability of creating a row of a certain weight d, i.e. a row with d non-zero elements. In a second random process d of the k elements in a row are then chosen uniformly at random without replacement to be non-zero. Finally, the non-zero elements are sampled uniformly at random from $\mathbb{F}_q \setminus \{0\}$.

The sparse random ensemble [SSV11, SLV11] is created by introducing a bias such that the probability of occurrence of the zero element is set to be higher than the probability of the other $\mathbb{F}_q$-elements. Depending on the bias, a more or less sparse random matrix is generated. For any entry $\mathsf{g}_{i,j}$ in $\mathbf{G}_\mathrm{T}$ let the following probabilities be defined as

$$P_0 \triangleq \Pr\{\mathsf{g}_{i,j} = 0\} \tag{2.10}$$

$$P_{\setminus 0} \triangleq \Pr\{\mathsf{g}_{i,j} \neq 0\} = 1 - P_0. \tag{2.11}$$

Moreover, the non-zero $\mathbb{F}_q$-elements occur with equal probability

$$\Pr\big\{\mathsf{g}_{i,j} = \alpha^i\big\} = \frac{P_{\setminus 0}}{q-1} \quad \text{for } \alpha^i \in \mathbb{F}_q \setminus \{0\}. \tag{2.12}$$

The row weight distribution of the sparse random ensemble is thus

$$\Omega(\xi) = \sum_{d=0}^{k} \binom{k}{d} P_{\setminus 0}^d \big(1 - P_{\setminus 0}\big)^{k-d} \xi^d. \tag{2.13}$$

In contrast to the previous ensembles, the random ensembles are preferably characterised by their density which amounts to $\Delta = 1 - 1/q$ for the standard random ensemble and to $\Delta = P_{\setminus 0}$ for the sparse random ensemble.

The Expurgated Random Ensembles

In both random ensembles, but particularly in the sparse random ensemble, all-zero rows occur with a non-zero probability. This is independent of whether the element-wise or the row-wise construction is used, since the two methods yield the same results if the respective row weight distributions (2.9) or (2.13) are used. As all-zero rows do not encode any information into encoded symbols, they are redundant and should be avoided.

The row-wise construction allows to adjust the row weight distributions such that all-zero rows cannot occur, i.e. by setting $\Omega_0 = 0$ and renormalising all other probabilities. The equivalent modification of the element-wise method is to discard all-zero rows as soon as they are created. The row weight distributions of these so-called expurgated random ensembles are

$$\Omega(\xi) = \frac{1}{q^k - 1} \sum_{d=1}^{k} \binom{k}{d} (q-1)^d \xi^d \tag{2.14}$$

for the expurgated standard random ensemble and

$$\Omega(\xi) = \frac{1}{1-(1-P_{\setminus 0})^k} \sum_{d=1}^{k} \binom{k}{d} P_{\setminus 0}^d \bigl(1-P_{\setminus 0}\bigr)^{k-d} \xi^d. \tag{2.15}$$

for the expurgated sparse random ensemble. Note that in the latter case $P_{\setminus 0}$ is the probability of sampling a non-zero element prior to removing the all-zero rows and thus $\Delta > P_{\setminus 0}$ now.

In terms of erasure correction both just introduced expurgated random ensembles perform excellently under ML decoding. The expurgated standard random ensemble is generally considered to be the optimal LT code ensemble under ML decoding, though a rigorous proof hereof is still missing. However, expurgation is usually not mentioned explicitly and thus only the standard random ensemble is discussed, since for not too small input sizes, the probability of all-zero rows is so small that the difference to the expurgated version is negligible. Accordingly, the expurgated sparse random ensemble is a near-optimal LT code ensemble under ML decoding given a constraint on the density Δ or equivalently on the average row weight $\bar{d}$. Their excellent performance has been verified in [SSV11, SLV11, SGV13] by means of Monte Carlo simulations and tight performance bounds. Further details will be provided in Chapter 3.

Concentrated Distributions

Concentrated degree or row weight distributions, which play an important role particularly in LDPC codes, can also be applied to LT codes. In the field of LDPC codes this type of row weight distributions is also known by the name right-regular distribution. The term *concentrated* hints at the sample space $\mathcal{D}$ that contains usually only one value or at most two neighbouring values, i.e. the row weight distribution has the form

$$\Omega(\xi) = \xi^d \qquad \text{or} \qquad \Omega(\xi) = \Omega_d \xi^d + \Omega_{d+1} \xi^{d+1}. \tag{2.16}$$

Ensembles arising from a row weight distribution with $|\mathcal{D}| = 1$ are denoted *purely concentrated* in this thesis.

Although ensembles from concentrated distributions can have a good erasure correction performance which can be almost indistinguishable from that of the sparse random ensemble [SSV11], one has to be aware of some issues before using them for an LT code ensemble:

1. A sample space $\mathcal{D}$ that contains only even values leads to binary LT ensembles that cannot be decoded. Therefore, a certain amount of odd values should be included to ensure that decoding matrices do not only have even

row weights. For concentrated ensembles with an (almost) even average row weight $\bar{d}$, where the sample space would usually consist only of $d = \bar{d}$, the odd row weights $d - 1$ and $d + 1$ should also be used.

2. Unless the average row weight is not in the range of one or two, concentrated ensembles have no rows of weight one. This means that BP decoding cannot even begin. So concentrated ensembles have to be entirely decoded by means of ML decoding. Moreover, ML decoding of concentrated ensembles is usually computationally more expensive than that of non-concentrated ones with the same average row weight $\bar{d}$. ML decoding involves pivoting by means of row additions. In general, to keep the number of non-zero entries in the decoding matrix as low as possible during the decoding process, it makes sense to use rows of low weight first, since they create the lowest number of new entries. Thereby a faster pivoting is facilitated and the weights of the initially high-weight rows get reduced before they are added to other rows. This effect is stronger in ensembles with a low average row weight.

The performance and the computational complexity of concentrated ensembles will be further examined and discussed in Section 3.5.3.

2.1.7 The Column Weight Distribution

According to [Sho06] the column weight distribution, i.e. the input node degree distribution, of an LT code generator matrix is fully determined by the usual parameters of an LT code ensemble. The probability that a particular (fixed but arbitrary) input node is connected to a particular output node of degree d is d/k, i.e. the probability that it is connected to a particular output node of arbitrary degree is $\sum_{d \in \mathcal{D}} \Omega_d d/k = \bar{d}/k$. As there are n independent output nodes, the probability that a particular input node is connected to i output nodes is $\binom{n}{i}\left(\frac{\bar{d}}{k}\right)^i\left(1-\frac{\bar{d}}{k}\right)^{n-i}$ and the column weight distribution is thus [Sho06]

$$\Xi(\xi) = \sum_{i=0}^{n} \binom{n}{i} \left(\frac{\bar{d}}{k}\right)^i \left(1 - \frac{\bar{d}}{k}\right)^{n-i} \xi^i. \tag{2.17}$$

The degree of an input node, or equivalently the weight of the corresponding column, is a good measure for its erasure resilience. For equally important input nodes it is thus desirable to obtain equal input node degrees, i.e. a concentrated distribution. In fact, the column weight distribution is usually controllable only in a rather limited way. While at the transmitter the edges of consecutively created output nodes can be connected to input nodes such that the input symbol degrees are fairly balanced, erasures on the channel weaken this concentration at the receiver.

For higher erasure probabilities, the so created LT codes appear to a receiver just like LT codes that have been created randomly according to a given row weight distribution. And by the usual random construction of LT code ensembles, a concentrated column weight distribution cannot be achieved.

2.1.8 Binary Images of Non-Binary Codes

So far, only the transmission over a symbol erasure channel (SEC) has been considered in this thesis, as it is usually assumed that symbols are erased completely. However, since in practice each $\mathbb{F}_q$-element or symbol is represented and transmitted in binary form, it may happen that even a single bit erasure leads to the erasure of a whole symbol. If bit erasures are spread over many $\mathbb{F}_q$-elements, it can have a very negative impact on the decodability of the received encoded symbols. Therefore, in corresponding scenarios, it can be useful to perform decoding on the binary equivalent of the non-binary code [SLV11].

The straightforward approach is to express each $\mathbb{F}_q$-symbol, be it a received symbol or an entry in the decoding matrix $\mathbf{G}_{\mathrm{R}}$, by a binary vector $\mathbf{a}^{(i)} = \left(a_0^{(i)}, a_1^{(i)}, \ldots, a_\mu^{(i)}\right)^{\mathsf{T}}$. The usual binary vector representation consists of the $\mu = \mathrm{ld}(q)$ binary coefficients of the polynomial that defines each non-zero $\mathbb{F}_q$-symbol, i.e. $\alpha^i \mod a(\alpha)$ with $i \in \{0, 1, \ldots, \mu-2\}$, where α is a primitive element of $\mathbb{F}_q$ and $a(\xi) = a_0 + a_1\xi + \ldots + a_{\mu-1}\xi^{\mu-1} + \xi^\mu$ is a primitive polynomial of $\mathbb{F}_q$ with coefficients from the prime subfield $\mathbb{F}_2$, i.e. $a_i \in \mathbb{F}_2$. Clearly, the zero element is represented by the binary all-zero vector. Besides the need for carrying out a modulus operation when multiplying $\mathbb{F}_q$-symbols in the binary vector representation, a more substantial drawback of this approach is that it is not clear how to deal with bit erasures within such binary vectors. And thus, additionally a different, more redundant representation of the $\mathbb{F}_q$-symbols is chosen in order to utilise also $\mathbb{F}_q$-symbols with bit erasures for decoding.

The Companion Matrix Representation

Each non-zero $\mathbb{F}_q$-element can be represented by a power of the so-called companion matrix [MS77]. The companion matrix corresponds to the primitive element α of $\mathbb{F}_q$ and in this context is defined as the $\mu \times \mu$ matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & a_0 \\ 1 & 0 & 0 & \cdots & 0 & a_1 \\ 0 & 1 & 0 & \ddots & 0 & a_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & a_{\mu-1} \end{pmatrix}. \tag{2.18}$$

The all-zero matrix of the same size corresponds to the zero element. The binary equivalent of the generator matrix $\mathbf{G}$ is obtained by exchanging each $\mathbb{F}_q$-symbol α^i, where $i \in \{0, 1, \ldots, q-2\}$, by the corresponding power of the companion matrix, i.e. by $\mathbf{A}^i$. In principle, the input symbols and the encoded symbols can be represented as matrices as well. However, that would be too redundant, so the binary vector representation is sufficient for the input and output symbols. Examples of the different representations of the elements in $\mathbb{F}_8$ are provided in Table 2.2. Moreover, the decoding of received symbols that contain bit erasures is briefly illustrated in the following.

Example 2.1. The information word $\mathbf{x} = \left(\alpha^4, \alpha^6\right)^\mathsf{T} \in \mathbb{F}_8^2$ (cf. Table 2.2) is encoded with a matrix $\mathbf{G}_\mathrm{T} \in \mathbb{F}_8^{2\times 3}$ and yields the encoded vector $\mathbf{y}_\mathrm{T} \in \mathbb{F}_8^3$:

$$\begin{aligned} \mathbf{y}_\mathrm{T} &= \mathbf{G}_\mathrm{T}\mathbf{x} \\ \Leftrightarrow \begin{pmatrix} \alpha^5 \\ \alpha^2 \\ 1 \end{pmatrix} &= \begin{pmatrix} \alpha^3 & \alpha^5 \\ 1 & \alpha^2 \\ \alpha^4 & \alpha^4 \end{pmatrix} \begin{pmatrix} \alpha^4 \\ \alpha^6 \end{pmatrix}. \end{aligned}$$

The vector $\mathbf{y}_\mathrm{T}$ is then transmitted over a *binary* erasure channel in its binary vector representation, i.e. as $(1, 1, 1, 0, 0, 1, 1, 0, 0)^\mathsf{T}$. On the channel three bit erasures occur, so that $\mathbf{y}_\mathrm{R}$ is given by $(1, 1, ?, ?, 0, ?, 1, 0, 0)^\mathsf{T}$. With a protocol discarding symbols with bit erasures, only one symbol would be left for decoding, which is certainly insufficient. However, the input symbols $\mathbf{x}$ can be recovered by using the equivalent binary vector and matrix representation, i.e.

$$\left(\begin{array}{c} 1 \\ 1 \\ \color{gray}? \\ \hline \color{gray}? \\ 0 \\ \color{gray}? \\ \hline 1 \\ 0 \\ 0 \end{array}\right) = \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ \color{gray}0 & \color{gray}1 & \color{gray}1 & \color{gray}1 & \color{gray}1 & \color{gray}0 \\ \hline \color{gray}1 & \color{gray}0 & \color{gray}0 & \color{gray}0 & \color{gray}1 & \color{gray}0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ \color{gray}0 & \color{gray}0 & \color{gray}1 & \color{gray}1 & \color{gray}0 & \color{gray}1 \\ \hline 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right) \left(\begin{array}{c} x_{\mathrm{B},1,0} \\ x_{\mathrm{B},1,1} \\ x_{\mathrm{B},1,2} \\ \hline x_{\mathrm{B},2,0} \\ x_{\mathrm{B},2,1} \\ x_{\mathrm{B},2,2} \end{array}\right),$$

and by solving the binary equivalent system of linear equations $\mathbf{y}_\mathrm{R} = \mathbf{G}_\mathrm{R}\mathbf{x}$ which results from the system above by removing the three erased bits from $\mathbf{y}_\mathrm{T}$ and the corresponding greyed out rows from the binary equivalent of the matrix $\mathbf{G}_\mathrm{T}$. The binary vector equivalent of the input symbols $\mathbf{x} = (x_1, x_2)^\mathsf{T}$ is given by $(x_{\mathrm{B},1,0}, x_{\mathrm{B},1,1}, \ldots, x_{\mathrm{B},2,2})^\mathsf{T}$. The grey lines in the above system of linear equations are merely drawn to allow for a better visual discrimination of the individual symbols. Since the matrix $\mathbf{G}_\mathrm{R}$ has full column rank, the above system can be solved completely, resulting in $\mathbf{x} = (0, 1, 1, 1, 0, 1)^\mathsf{T}$, which is the binary equivalent of the original information word.

The Density of the Binary Matrix Equivalent

By transforming a non-binary code into its binary image, the characteristics of the code change as well [SLV11]: most importantly, the row weight distribution is not preserved. Also the density decreases, while the average row weight increases (slightly, depending on μ). As the non-zero $\mathbb{F}_q$-symbols occur with equal probability, the expected density of their binary matrix equivalents amounts to

$$\Delta_{\mathrm{S}\to\mathrm{B}} = \frac{2^{\mu-1}}{2^{\mu}-1} \xrightarrow{\mu\to\infty} \frac{1}{2}.$$

Then, the binary image of a non-binary matrix with $\mathbb{F}_q$-symbol density Δ_S has the density $\Delta_\mathrm{B} = \Delta_{\mathrm{S}\to\mathrm{B}} \cdot \Delta_\mathrm{S}$ with $\Delta_\mathrm{B} < \Delta_\mathrm{S}$, since $\Delta_{\mathrm{S}\to\mathrm{B}} < 1$ if $\mu > 1$.

The derivation of the above stated expected density $\Delta_{\mathrm{S}\to\mathrm{B}}$ can best be explained by means of the exemplary field $\mathbb{F}_8$, whose companion matrix representation is provided in Table 2.2. Considering all but the zero-matrix, it can be observed that in a fixed but arbitrary row (or column) among all matrices, each of the $2^\mu - 1 = 7$ possible binary non-zero patterns of length $\mu = 3$ occurs exactly once, so the total number of ones in a particular row or column amounts to $\sum_{i=1}^{\mu} \binom{\mu}{i} i = \mu 2^{\mu-1}$. Finally, to obtain the density, this number has to be normalised to one entry instead of a row or column and also only to one matrix instead of to all but the zero-matrix. Therefore, the number $\mu 2^{\mu-1}$ is divided by μ, i.e. the number of entries per row or column and $2^\mu - 1 = 7$, the number of non-zero matrices, which yields the stated expression for $\Delta_{\mathrm{S}\to\mathrm{B}}$.

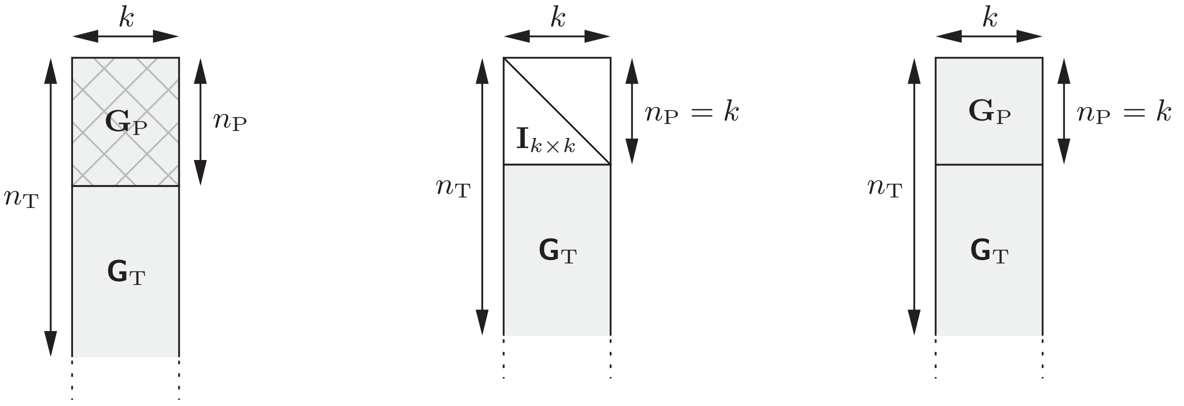
As the average number of ones per row of the binary matrix equivalents is greater than one, the average row weight $\bar{d}_\mathrm{B} = \mu\, \Delta_{\mathrm{S}\to\mathrm{B}}\, \bar{d}_\mathrm{S}$ of the equivalent binary code is greater than the average row weight $\bar{d}_\mathrm{S}$ of the q-ary code. Despite binary operations being computationally less complex than q-ary operations, the computational cost of solving a larger binary equivalent system of linear equations, which even has a higher average row weight, outweighs the savings of less complex operations by far. So, transforming a non-binary code to its binary image comes at the price of an increased decoding complexity. Nevertheless, the binary image allows partially received $\mathbb{F}_q$-symbols that contain bit erasures to be used for decoding.

2.2 Structured LT Code Ensembles

General LT code ensembles are created via several random processes which do not guarantee that for instance the first k rows are linearly independent. Particularly

Table 2.2: Different representations of $\mathbb{F}_8$ as defined by the primitive polynomial $a(\xi) = \xi^3 + \xi + 1$. Note that the binary vector representation is equal to the first column of the respective companion matrix representation.

exponential representation α^i	polynomial representation $\alpha^i \mod \left(\alpha^3 + \alpha + 1\right)$	binary vector representation $\left(a_0^{(i)},\, a_1^{(i)},\, a_2^{(i)}\right)^\mathsf{T}$	companion matrix representation $\mathbf{A}^i$
0	0	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
1	1	$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
α	α	$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
α^2	α^2	$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$
α^3	$\alpha + 1$	$\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$
α^4	$\alpha^2 + \alpha$	$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$
α^5	$\alpha^2 + \alpha + 1$	$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$
α^6	$\alpha^2 \qquad + 1$	$\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$



$$\mathbf{G}_{\mathrm{struct},\mathrm{T}} = \begin{pmatrix} \mathbf{G}_\mathrm{P} \\ \mathbf{G}_\mathrm{T} \end{pmatrix}$$

$\mathbf{G}_\mathrm{P}$ created offline, full column rank

$\mathbf{G}_\mathrm{T} \sim \Omega(\xi)$

(a) A generic structured LT code ensemble.

$$\mathbf{G}_{\mathrm{sys},\mathrm{T}} = \begin{pmatrix} \mathbf{I}_{k\times k} \\ \mathbf{G}_\mathrm{T} \end{pmatrix}$$

$\mathbf{G}_\mathrm{P} = \mathbf{I}_{k\times k}$

$\mathbf{G}_\mathrm{T} \sim \Omega(\xi)$

(b) A conventionally systematic LT code ensemble.

$$\mathbf{G}_{\mathrm{sys},\mathrm{T}} = \begin{pmatrix} \mathbf{G}_\mathrm{P} \\ \mathbf{G}_\mathrm{T} \end{pmatrix}$$

$\mathbf{G}_\mathrm{P} \sim \Omega(\xi)$, created offline, invertible, for BP decoding triangularisable

$\mathbf{G}_\mathrm{T} \sim \Omega(\xi)$

(c) A constructed systematic LT code ensemble.

Figure 2.8: Structured LT code ensembles.

for users with reliable channels, this is wasteful and can be resolved by allowing only certain matrices, e.g. only matrices of rank k, to constitute the first part, i.e. the prefix of an LT code.

A prefix matrix $\mathbf{G}_\mathrm{P}$ of height n_P will be denoted the n_P-prefix of an LT code ensemble. In the following, different structured prefixes will be discussed that can be attached to LT code ensembles as depicted in Figure 2.8(a). Since the choice or construction of appropriate matrices can be more involved, the prefix matrix $\mathbf{G}_\mathrm{P}$ is generally determined offline and is kept constant for all transmissions.

The term "structured" can be understood in a very general sense. It may not only be applied to matrices that have a clearly visible structure such as a diagonal matrix, but also to matrices with more subtle structures like the property of full column rank, the MDS property [BL11,BGL13] or particular row or column weight profiles. Here, the row or column weight profile is understood as the distribution of the actual row or column weight, i.e. it counts the rows or columns of weight d of a particular matrix instantiation.

Usually, a code or an ensemble is considered "universal" if the erasure correction performance is good independently from the channel realisation. Unstructured or plain LT code ensembles are therefore universal on the erasure channel. However, in this strict sense, structured ensembles are not universal, since even an ensemble with a reasonably structured prefix is supposed to yield a different – though better – erasure correction performance than without such a prefix, particularly for good channels. In this thesis, the term “universal” will be used in a broader sense, in that a structured ensemble is also called universal if its erasure correction performance is at least equal to or even better than that of the unstructured ensemble for any erasure probability on the channel.

2.2.1 Conventionally Systematic LT Code Ensembles

In many communication systems the use of systematic codes is favoured over non-systematic codes, since in good channel conditions the message can often be obtained from the systematic part, without requiring any decoding steps at all. And even if a few erasures occur on the channel, a systematic code may enable a less complex decoding, preferably also at a lower reception overhead than a non-systematic one.

The usual approach however, i.e. simply using the identity matrix $\mathbf{I}_{k\times k}$ as k-prefix, as sketched in Figure 2.8(b), does not necessarily deliver better results than the plain LT ensemble, i.e. without this prefix. As described in [SL05, Sho06, SL11], the decoding failure probability depends upon the number of systematic symbols among the received encoded symbols and thereby on the channel quality. Also it is claimed that the decoding properties of such a system become particularly bad if the number of received systematic symbols among all received symbols is small, i.e. that a conventionally systematic LT ensemble is not universal. While the dependence of the decoding failure probability on the channel quality is undoubted, the latter claim does not hold for all LT ensembles as will be shown in Chapter 4.

2.2.2 The Systematic LT Code Construction

As the performance of conventionally systematic ensembles was considered too low in [SL05,Sho06,SL11], the so-called systematic construction was proposed instead. First, some prerequisites as well as the encoding and decoding are described and then, methods for creating suitable prefix matrices are revised or introduced.

Constraints on the Prefix Matrix

The prefix matrix for the constructed systematic LT code ensemble is a full rank $k \times k$ matrix and its row weight profile should be a typical representative of the row weight distribution $\Omega(\xi)$. By the latter property it is ensured that the graph is self-similar no matter which $k(1+\varepsilon_\mathrm{R})$-subset of the output symbols is received, i.e. $\mathbf{G}_\mathrm{R} \sim \Omega(\xi)$ almost independently of the erasure probability on the channel. Only for very good channels, where the systematic part has a high probability to be received as a whole, the performance is expected to significantly exceed that of the plain ensemble. Depending on the designated decoding algorithm, BP or ML decoding, $\Omega(\xi)$ has to be chosen or designed accordingly. For BP decoding, the k-prefix additionally has to be permutation-triangularisable, i.e. it has to be transformable into an upper (or lower) triangular matrix just by the multiplication with a $k \times k$ permutation matrix.

Systematic Encoding and Decoding

In the usual LT coding setup, the input nodes or input symbols $\mathbf{x}$ are equated with the data $\mathbf{s}$ that has to be transmitted and the terms are consequently used synonymously, except in this section. In the systematic LT code construction, which was proposed in [SL05, Sho06, SL11], the first k *output* symbols are now equated with the data symbols such that they are now systematically contained in the encoded symbols, i.e. $\mathbf{y}_{\mathrm{T}} = (y_1, y_2, \ldots, y_{n_{\mathrm{T}}})^{\mathsf{T}} = (s_1, s_2, \ldots, s_k, y_{k+1}, \ldots, y_{n_{\mathrm{T}}})^{\mathsf{T}}$. For the determination of the following (non-systematic) output symbols y_{k+1} to $y_{n_{\mathrm{T}}}$ the symbols $\mathbf{x}$ are required which have to be calculated from the systematic output symbols by solving $\mathbf{G}_{\mathrm{P}}\mathbf{x} = \mathbf{s}$. Then, the symbols $\mathbf{x}$ can be used to generate the non-systematic output symbols by $\mathbf{G}_{\mathrm{T}}\mathbf{x} = (y_{k+1}, \ldots, y_{n_{\mathrm{T}}})^{\mathsf{T}}$.

On the receiver side, either all systematic output symbols are received, in that case no decoding has to be performed, or a mixture of systematic and non-systematic output symbols are received. In the latter case, the input symbols have to be recovered first, and then by an additional encoding step using the k-prefix matrix $\mathbf{G}_{\mathrm{P}}$ the missing data symbols are recovered.

Constructing the Systematic Prefix for ML Decoding

The rationale of the systematic construction besides the obvious systematic transmission is that the decoding matrix is self-similar in order to achieve quasi-independence from the channel quality, i.e. universality in the aforementioned sense. Thus, the k rows of the prefix matrix are created just like all others according to $\Omega(\xi)$. However, before actually using this k-prefix, it is checked for full rank. If it does not have full rank, it is discarded and a new k-prefix is sampled.

When using a relatively high field order q and a row weight distribution with sufficiently high average row weight $\bar{d}$, the probability of directly sampling a matrix with full rank is reasonably high, but for lower field sizes or lower average row weight it is very likely that multiple trials are required to obtain such a matrix. While this is the method of choice for creating ML decodable prefixes, there are smarter construction algorithms for BP decodable prefixes.

Constructing the Systematic Prefix for BP Decoding

A construction method for BP decodable prefixes has been proposed in [SL05, Sho06, SL11] and is briefly revised here. Although it is originally formulated for binary codes, it directly applies to non-binary codes as well. By this approach, initially a slightly taller $k(1+\varepsilon)$-prefix $\widetilde{\mathbf{G}}_{\mathrm{P}} \sim \Omega(\xi)$ is generated. A counter j is initialised with zero and an auxiliary matrix $\widetilde{\mathbf{G}}_{\mathrm{P,aux}} \triangleq \widetilde{\mathbf{G}}_{\mathrm{P}}$ is defined. Then, the following steps are performed as long as $j < k$:

1. Find a row of weight 1 in the auxiliary matrix $\widetilde{\mathbf{G}}_{\mathrm{P,aux}}$; if none exists, return an error and stop; otherwise, set i_j to be equal to the index of the corresponding row in $\widetilde{\mathbf{G}}_{\mathrm{P}}$.
2. Find the unique non-zero position of this row, remove the column corresponding to that position from the auxiliary matrix $\widetilde{\mathbf{G}}_{\mathrm{P,aux}}$ and increase j by one.

The final k-prefix $\mathbf{G}_{\mathrm{P}}$ is obtained as the rows $i_1, \ldots, i_k$ from $\widetilde{\mathbf{G}}_{\mathrm{P}}$. Note that $\mathbf{G}_{\mathrm{P}}$ is a permutation-triangularisable matrix which implicitly has full column rank. Its row weight profile should be very similar to the expected row weight distribution $k\Omega(\xi)$. Differences may occur due to non-zero variances of the random processes or result from the selection of k suited rows out of $k(1+\varepsilon)$ rows, since usually $k\Omega(\xi)$ has non-integer coefficients.

An Efficient Systematic Prefix Construction for BP Decoding

A necessary condition for BP decodability given a $k \times k$ matrix is that it has full rank, while a sufficient condition is that the matrix is permutation-triangularisable. Note that this sufficient condition already implies the necessary condition, since a triangular matrix always has full rank.

It is assumed that the row weight distribution $\Omega(\xi)$ generates a BP decodable ensemble with high probability. Now, instead of the previously described approach from [SL05, Sho06, SL11] of randomly generating $k(1+\varepsilon)$ rows according to $\Omega(\xi)$ and selecting only a BP decodable k-subset, the following new method enables to create a full rank prefix matrix directly without trial and error, which is BP decodable and is a typical representative of $\Omega(\xi)$. The latter is indeed the weakest criterion and may be revised as far as to allow the other two essential conditions to be met.

To this end, the coefficients $k\Omega_d$ of $k\Omega(\xi)$ are rounded towards the nearest integer. Then, $\mathrm{rnd}(k\Omega_d)$ rows of weight d are created. The lowermost $\mathrm{rnd}(k\Omega_1)$ rows are assigned weight 1, the $\mathrm{rnd}(k\Omega_2)$ rows above these ones are assigned weight 2 and

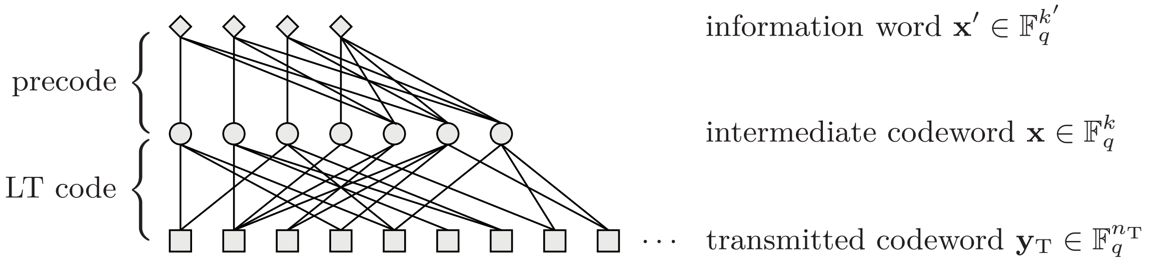

precode
LT code
· · ·
information word $\mathbf{x}' \in \mathbb{F}_q^{k'}$
intermediate codeword $\mathbf{x} \in \mathbb{F}_q^{k}$
transmitted codeword $\mathbf{y}_\mathrm{T} \in \mathbb{F}_q^{n_\mathrm{T}}$

Figure 2.9: An exemplary Raptor encoding graph. The depicted Raptor code is a concatenation of the $(7,4)$ Hamming code and an LT code.

so forth. In the next step, all main diagonal entries are sampled from $\mathbb{F}_q \setminus \{0\}$. And finally, in each row i, choose uniformly at random $d-1$ entries from the $i-1$ possible entries on the right-hand side of the main diagonal and assign randomly chosen elements from $\mathbb{F}_q \setminus \{0\}$ to these $d-1$ entries. In each row i the assigned weight d has to fulfil the condition $1 \leq d \leq i$. If this condition is not met, i.e. if $d > i$, the weight d of row i has to be reduced to the maximum row weight that is smaller than or equal to i and has a positive probability of occurrence, even if the resulting row weight profile deviates from the expected row weight profile with rounded coefficients.

2.3 Raptor Codes

The commercial name Raptor code [Sho04, Sho06], a portmanteau derived from the name “rapid Tornado code”, denotes a concatenation of a rateless component such as an LT code and one or more stages of high-rate precodes. Classical Tornado codes [LMS$^+$97] are a class of BP decodable erasure-resilient codes based on irregular bipartite graphs. Their development has led to the insight that irregular degree distributions are essential for constructing capacity achieving codes, which influenced also the design of LDPC codes.

An example of a Raptor encoding graph is depicted in Figure 2.9. The information word $\mathbf{x}'$ is first encoded by a precode, for example a Hamming code or an LDPC code. The obtained intermediate codeword $\mathbf{x}$ is then further encoded by an LT code. This construction allows to successfully decode an information word, even if the LT code cannot recover all intermediate nodes.

As already mentioned, LT codes like their fixed rate relatives, i.e. LDGM codes, suffer from a high erasure floor. The quantity that mostly influences the erasure floor is the average row weight $\bar{d}$ or equivalently the density Δ of the LT code matrix. However, increasing $\bar{d}$ comes at the cost of a higher computational complexity, and it should thus rather be avoided if a certain code rate loss is affordable, which is due to the precode.

Instead, LT codes should be kept sparse and should only be seen as the component that introduces ratelessness in a concatenated scheme and that recovers a large fraction of the intermediate nodes, but not necessarily all. The erasure floor of a properly designed LT code is thus dominated by a small fraction of undecodable LT code input nodes, i.e. intermediate nodes, per information word. The task of lowering the erasure floor is left to one or more high-rate precodes. A suitable precode, i.e. one that can (almost) guarantee the correction of a few erasures per codeword, can lower the erasure floor more efficiently than a single LT code with a higher average row weight. More details on design and analysis of finite length Raptor codes will be provided in Chapter 6.

Chapter 3

Finite Length Analysis under Optimal Erasure Decoding

Provided a transmission over a symbol erasure channel (SEC), ML decoding of an LT code ensemble $\mathfrak{L}$ corresponds to solving a consistent system of n_{R} random linear equations in k unknowns over a finite field $\mathbb{F}_q$. The probability that the system is solvable is equal to the probability that the decoding matrix $\mathbf{G}_{\mathrm{R}}$ at the receiver has rank k. Hence, the word erasure probability $P^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right)$ after ML decoding equals the probability that $\mathbf{G}_{\mathrm{R}}$ has not rank k. An information word is considered erased if at least one information symbol remains erased after decoding. Apart from the word erasure probability $P^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right)$ also the symbol erasure probability $P^{[\mathfrak{L}]}\left(\cancel{\mathcal{S}}\right)$ is used to assess the erasure correction performance of a code and its suitability for precoding.

A vast amount of publications exists (e.g. [Lan93, ER63, Odl81, Kol94, Cal96, Cal97, BKW97, Coo00b, Coo00a]) that examine rank properties of random matrices. Especially in the former Soviet Union many researchers have extensively dealt with this topic, whose results, being published in Russian, have long been unknown to most other researchers. A survey of many results is provided, e.g. in [Kol99] and [Lev05]. Nevertheless, all contain certain restrictions on either the randomness or on the dimensions of the matrix. Restrictions in terms of randomness are for instance that only uniform randomness is considered, i.e. each entry of the random matrix is sampled uniformly from the set of $\mathbb{F}_q$-elements (standard random ensemble) or uniformly from the set of non-zero $\mathbb{F}_q$-elements and with an increased probability of occurrence of the zero element (sparse random ensemble). Restrictions in terms of the matrix dimensions are, e.g. considering only square matrices or making only asymptotic statements.

However, for the analysis and the design of random linear codes, e.g. fountain codes or network codes, with a finite, particularly a small or medium, information word length and a given row weight distribution, i.e. an output node degree distribution, the results from the literature are not sufficient. For the major part of this thesis, the following two questions will be dealt with in great detail for different row weight distributions:

1. What is the probability that a random $(n \times k)$ matrix over a finite field $\mathbb{F}_q$ does not have full column rank? In the following, $n \geq k$ is assumed.

2. What is the probability that a system of n linear equations with random coefficients from $\mathbb{F}_q$ cannot be solved for an arbitrary but fixed unknown?

In the setup of LT code ensembles the first probability is equivalent to the residual word erasure probability $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\big)$ after ML decoding. The second probability is equivalent to the residual symbol erasure probability $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{S}}\big)$ after ML decoding. In the literature, no exact expressions are known for these probabilities other than for $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\big)$ in the very specific case of uniform randomness, i.e. for the standard random ensemble. In Section 3.2, this probability is briefly reviewed and a new exact expression is determined for the expurgated random ensemble.

However, in the general case, we have to content ourselves with upper and lower bounds on these probabilities. In contrast to the residual erasure probabilities, the corresponding bounds are marked by over- or underlining the character P, e.g. a lower bound on symbol level is denoted $\underline{P}^{[\mathfrak{L}]}\big(\cancel{\mathcal{S}}\big)$. But before starting with the derivation of such bounds, some basic terms have to be defined or reviewed.

3.1 Some Basics and Definitions

Definition 3.1. The rank of a matrix $\mathbf{G} \in \mathbb{F}_q^{n \times k}$ is the number of linearly independent rows or columns of $\mathbf{G}$ and is denoted $\operatorname{rank}(\mathbf{G})$. It is equal to the image dimension of $\mathbf{G}$, i.e. $\operatorname{rank}(\mathbf{G}) = \dim(\operatorname{img}(\mathbf{G}))$.

A matrix $\mathbf{G} \in \mathbb{F}_q^{n \times k}$ is said to have full column rank if $\operatorname{rank}(\mathbf{G}) = k$.

Definition 3.2. The kernel (or null space) of a matrix $\mathbf{G} \in \mathbb{F}_q^{n \times k}$ is the set of all vectors $\mathbf{x} \in \mathbb{F}_q^k$ that map to zero

$$\ker(\mathbf{G}) = \Big\{\mathbf{x} \in \mathbb{F}_q^k \;:\; \mathbf{G}\mathbf{x} = \mathbf{0}\Big\}. \tag{3.1}$$

Definition 3.3. The nullity (or defect) of a matrix $\mathbf{G} \in \mathbb{F}_q^{n \times k}$ is equal to the kernel dimension, i.e. $\operatorname{nullity}(\mathbf{G}) = \dim(\ker(\mathbf{G}))$.

In other words, the nullity of $\mathbf{G}$ is equal to the maximum number of columns that can be removed from $\mathbf{G}$ such that the rank does not change. The removed columns are then linearly dependent on the remaining columns in the matrix.

Theorem 3.4 (Rank-nullity theorem)**.** Given a matrix $\mathbf{G} \in \mathbb{F}_q^{n \times k}$, then $\operatorname{rank}(\mathbf{G}) + \operatorname{nullity}(\mathbf{G}) = k$.

Krawtchouk Polynomials

Definition 3.5. The Krawtchouk polynomial [MS77] is defined as

$$\mathcal{K}_\varsigma(\xi;\nu) = \sum_{i=0}^{\varsigma} (-1)^i (q-1)^{\varsigma-i} \binom{\xi}{i} \binom{\nu-\xi}{\varsigma-i}, \quad \varsigma = 0, 1, \ldots, \nu, \tag{3.2}$$

for any positive integer ν and prime power q. The latter usually corresponds to the respective field order.

Corollary 3.6. The Krawtchouk polynomial $\mathcal{K}_\varsigma(\xi;\nu)$ for $\xi = 0$ is

$$\mathcal{K}_\varsigma(0;\nu) = (q-1)^\varsigma \binom{\nu}{\varsigma} \tag{3.3}$$

which can be obtained by using the fact that $\binom{\xi}{i} > 0$ if $\xi, i \in \mathbb{N}_0$ and $0 \leq i \leq \xi$, but that $\binom{\xi}{i} = 0$ in all other cases.

q-Analogues

Definition 3.7. The q-analogue of a number $\nu \in \mathbb{N}$, also denoted q-number or q-bracket of ν, is

$$[\nu]_q = \frac{q^\nu - 1}{q - 1}. \tag{3.4}$$

Definition 3.8. The q-analogue of the factorial, usually denoted q-factorial is

$$\begin{aligned} [\nu]_q! &= [1]_q \cdot [2]_q \cdot \ldots \cdot [\nu-1]_q \cdot [\nu]_q & (3.5) \\ &= \frac{q-1}{q-1} \cdot \frac{q^2-1}{q-1} \cdot \ldots \cdot \frac{q^{\nu-1}-1}{q-1} \cdot \frac{q^\nu - 1}{q-1} & (3.6) \end{aligned}$$

Definition 3.9. The q-binomial coefficients, also denoted Gaussian coefficients are given by

$$\begin{aligned} \begin{bmatrix} \nu \\ \varsigma \end{bmatrix}_q &= \frac{[\nu]_q!}{[\nu-\varsigma]_q! [\varsigma]_q!} & (3.7) \\ &= \frac{(q^\nu - 1)(q^\nu - q)\left(q^\nu - q^2\right) \cdot \ldots \cdot \left(q^\nu - q^{\varsigma-1}\right)}{(q^\varsigma - 1)(q^\varsigma - q)(q^\varsigma - q^2) \cdot \ldots \cdot (q^\varsigma - q^{\varsigma-1})} & (3.8) \end{aligned}$$

and determine the number of ς-dimensional subspaces of the ν-dimensional vector space $\mathbb{F}_q^\nu$.

Note that the number of ordered ς-tuples of linearly independent vectors in $\mathbb{F}_q^\nu$ is

$$(q^\nu - 1)(q^\nu - q)\big(q^\nu - q^2\big)\dots\big(q^\nu - q^{\varsigma-1}\big). \tag{3.9}$$

The first vector can be chosen from the set of non-zero vectors, while the second vector is chosen from the set of non-zero vectors that are no multiples of the first one. In general, each vector is chosen from the set of non-zero vectors that does not contain the span of the preceding vectors.

3.2 Word Erasure Probabilities of Random Ensembles

3.2.1 The Standard Random Ensemble

The standard random ensemble is by far the best examined random matrix. Some results, e.g. on the rank profile, i.e. the rank distribution in an ensemble, date back to the work of Landsberg [Lan93] in 1893:

Theorem 3.10. The number of $(n_\mathrm{R} \times k)$ matrices in $\mathbb{F}_q$ of rank r is

$$N(n_\mathrm{R} \times k,\, r) = \begin{bmatrix} n_\mathrm{R} \\ r \end{bmatrix}_q \big(q^k - 1\big)\big(q^k - q\big)\big(q^k - q^2\big) \cdot \ldots \cdot \big(q^k - q^{r-1}\big). \tag{3.10}$$

Proof. There are $\left[\begin{smallmatrix} n_\mathrm{R} \\ r \end{smallmatrix}\right]_q$ r-dimensional subspaces of $\mathbb{F}_q^{n_\mathrm{R}}$. And each subspace is spanned by r linearly independent vectors in $\mathbb{F}_q^k$ of which there are $\big(q^k - 1\big)\big(q^k - q\big)\big(q^k - q^2\big) \cdot \ldots \cdot \big(q^k - q^{r-1}\big)$ possibilities. □

A special case of the above theorem is particularly useful as it leads directly to the (known) exact word erasure probability $P^{[\mathfrak{L}]}\big(\cancel{W}\big)$ of the standard random ensemble, given that n_R encoded symbols have been received.

Corollary 3.11. The number of $(n_\mathrm{R} \times k)$ matrices with full column rank (i.e. rank k) is (cf. (3.9))

$$N(n_\mathrm{R} \times k,\, r = k) = \prod_{i=0}^{k-1} \big(q^{n_\mathrm{R}} - q^i\big). \tag{3.11}$$

Corollary 3.12. The residual word erasure probability of the standard random ensemble after ML decoding is

$$P^{[\mathfrak{L}]}\big(\cancel{W}\big) = 1 - \frac{N(n_\mathrm{R} \times k,\, r = k)}{\sum_{r=0}^{k} N(n_\mathrm{R} \times k,\, r)},$$

where the denominator comprises the total number of $(n_\mathrm{R} \times k)$ matrices

$$\sum_{r=0}^{k} N(n_\mathrm{R} \times k,\, r) = q^{k n_\mathrm{R}} \tag{3.12}$$

which yields (cf. e.g. [LNC97])

$$P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\big) = 1 - \frac{\prod_{i=0}^{k-1} \left(q^{n_\mathrm{R}} - q^{i}\right)}{q^{k n_\mathrm{R}}} = 1 - \prod_{i=n_\mathrm{R}-k+1}^{n_\mathrm{R}} \left(1 - q^{-i}\right). \tag{3.13}$$

3.2.2 The Expurgated Random Ensemble

The expurgated random ensemble is generated from the standard random ensemble by removing all-zero rows as soon as they are created and replacing them with newly sampled non-zero rows.

Theorem 3.13. The residual word erasure probability after ML decoding of the expurgated random ensemble amounts to

$$P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\big) = 1 - \frac{\prod_{i=0}^{k-1} \left(q^{n_\mathrm{R}} - q^{i}\right) - \sum_{i=1}^{n_\mathrm{R}-k} \binom{n_\mathrm{R}}{i} N_{\backslash\mathbf{0}}((n_\mathrm{R}-i) \times k,\, r=k)}{\left(q^{k}-1\right)^{n_\mathrm{R}}}, \tag{3.14}$$

where $N_{\backslash\mathbf{0}}((n_\mathrm{R}-i) \times k,\, r=k)$ is the number of $((n_\mathrm{R}-i) \times k)$ matrices of rank k that do not contain any all-zero rows.

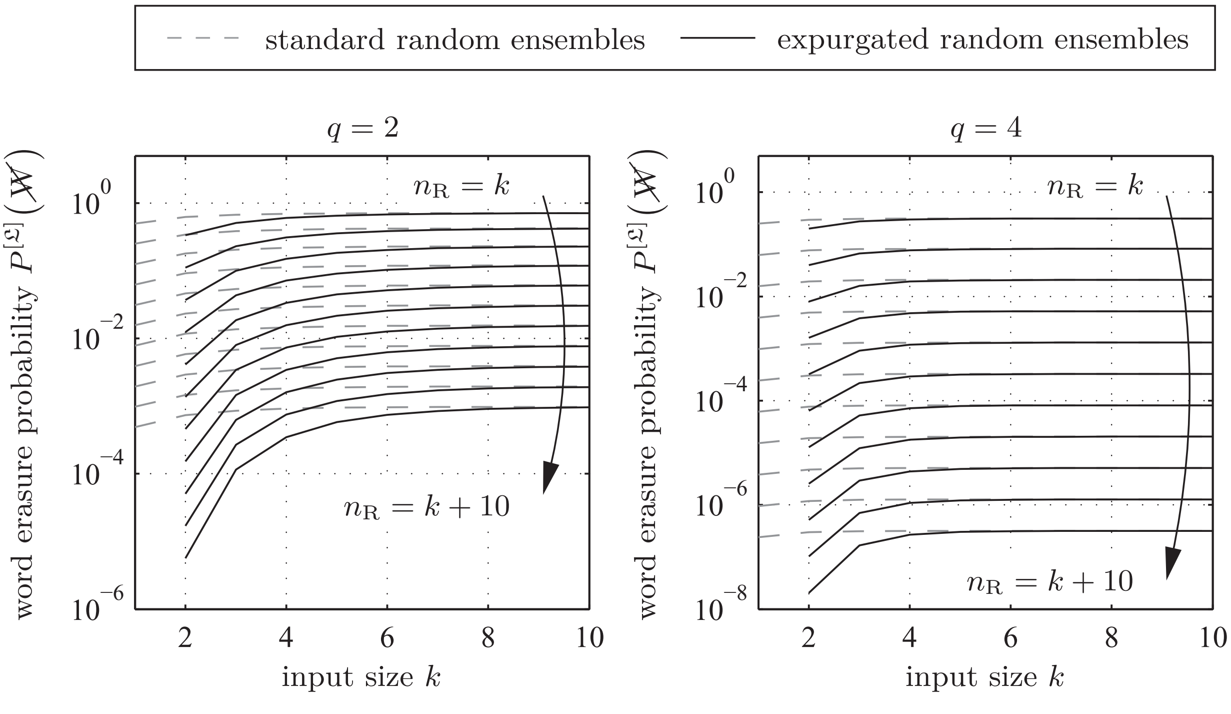
Proof. The number of all $(n_\mathrm{R} \times k)$ matrices not containing any all-zero rows is

$$\sum_{r=0}^{k} N_{\backslash\mathbf{0}}(n_\mathrm{R} \times k,\, r) = \left(q^{k}-1\right)^{n_\mathrm{R}}, \tag{3.15}$$

while the number of $(n_\mathrm{R} \times k)$ matrices with rank k has to be calculated recursively

$$N_{\backslash\mathbf{0}}(n_\mathrm{R} \times k,\, r=k) = N(n_\mathrm{R} \times k,\, r=k) - \sum_{i=1}^{n_\mathrm{R}-k} \binom{n_\mathrm{R}}{i} N_{\backslash\mathbf{0}}((n_\mathrm{R}-i) \times k,\, r=k) \tag{3.16}$$

by subtracting the number of rank k matrices that contain $i \in \{1, \dots, n_\mathrm{R}-k\}$ all-zero rows from the total number $N(n_\mathrm{R} \times k,\, r=k)$ of rank k matrices. The subtrahends thus constitute the numbers of rank k matrices of size $((n_\mathrm{R}-i) \times k)$ matrices without all-zero rows, multiplied by $\binom{n_\mathrm{R}}{i}$ to count the number of ways



(a) (b)

Figure 3.1: Word erasure probabilities of standard and expurgated random ensembles according to (3.13) and (3.14), respectively, for absolute symbol reception overheads from 0 to 10 over $\mathbb{F}_2$ and $\mathbb{F}_4$ for very small input sizes. For $k = 1$ the expurgated ensemble achieves $P^{[\mathfrak{L}]}\left(\cancel{W}\right) = 0$, since it is essentially a repetition code, which is the only – though trivial – rateless MDS code.

that i all-zero rows can be inserted into these matrices. The recursion ends with $(k \times k)$ matrices, since the number of square full rank matrices is the same in both ensembles, i.e. $N_{\setminus\mathbf{0}}(k \times k,\, r = k) = N(k \times k,\, r = k)$. Finally, the exact probability

$$P^{[\mathfrak{L}]}\left(\cancel{W}\right) = 1 - \frac{N_{\setminus\mathbf{0}}(n_\mathrm{R} \times k,\, r = k)}{\sum_{r=0}^{k} N_{\setminus\mathbf{0}}(n_\mathrm{R} \times k,\, r)} \tag{3.17}$$

of choosing a matrix of rank less than k from the expurgated ensemble can be determined from (3.11), (3.15) and (3.16). □

The residual word erasure probabilities of corresponding standard and expurgated random ensembles are plotted in Figure 3.1. For very small input sizes and field orders there exists a relevant difference in the erasure correction capabilities of the two ensembles. With increasing input sizes, however, the probability of occurrence of all-zero rows quickly converges to zero and the difference between the two types of ensembles vanishes. Thus, the word erasure probability of the standard random ensemble is a sufficiently good approximation for that of the expurgated ensemble.

3.3 Bounds on the Word and Symbol Erasure Probability

As already mentioned at the beginning of this chapter, the exact residual erasure probabilities under ML decoding on word and symbol level are not known in general. Only the residual word erasure probabilities of the standard and the expurgated random ensemble could be determined exactly as shown in the previous section. Consequently, upper and lower bounds on the respective probabilities are required in order to assess the erasure correction performance of general LT code ensembles. Such a set of four general bounds is presented subsequently.

3.3.1 An Upper Bound on the Word Erasure Probability

Theorem 3.14 (from [SGV13]). Given an LT code ensemble $\mathfrak{L} = \left(\mathbf{G}_{\mathrm{R}} \sim \Omega(\xi),\, \mathbb{F}_q^{n_{\mathrm{R}} \times k}\right)$, an upper bound on the word erasure probability $P^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right)$ after ML decoding is[1]

$$\overline{P}^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right) = \sum_{w=1}^{k} \binom{k}{w} (q-1)^{w-1} \left[\frac{1}{q} \sum_{d \in \mathcal{D}} \Omega_d \frac{\sum_{l=0}^{d} \binom{w}{l}\binom{k-w}{d-l}\left[1-(1-q)^{1-l}\right]}{\binom{k}{d}} \right]^{k\gamma_{\mathrm{R}}} \tag{3.18}$$

$$= \sum_{w=1}^{k} \binom{k}{w} (q-1)^{w-1} \left[\frac{1}{q} + \frac{q-1}{q} \sum_{d \in \mathcal{D}} \Omega_d \cdot \frac{\mathcal{K}_d(w;k)}{\mathcal{K}_d(0;k)} \right]^{k\gamma_{\mathrm{R}}} \tag{3.19}$$

with the inverse reception code rate $\gamma_{\mathrm{R}} = 1 + \varepsilon_{\mathrm{R}}$. The second, more compact variant comprises the well-known Krawtchouk polynomial (cf. Definition 3.5).

Proof. The probability $P^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right)$ is equal to the probability that $\mathbf{G}_{\mathrm{R}}$ does not have full column rank

$$P^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right) = \Pr\{\mathrm{rank}(\mathbf{G}_{\mathrm{R}}) < k\}, \tag{3.20}$$

i.e. the probability that the kernel of $\mathbf{G}_{\mathrm{R}}$ is non-trivial

$$P^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right) = \Pr\{\exists \mathbf{x} \in \ker(\mathbf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}\}. \tag{3.21}$$

This is equivalent to the probability that an arbitrary information word cannot be uniquely determined, since the solution of $\mathbf{G}_{\mathrm{R}}\mathbf{x} = \mathbf{y}_{\mathrm{R}}$ is a $(k - \mathrm{rank}(\mathbf{G}_{\mathrm{R}}))$-dimensional vector space. The expression in (3.21) $P^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right)$ denotes the probability that the non-trivial kernel $\ker(\mathbf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}$ is not empty, which can be equivalently

[1] For notational convenience it is implied that probabilities and their bounds are limited from above by one. The operation $\min\{1, \cdot\}$ is omitted.

rephrased as the probability that its cardinality is greater than or equal to one:

$$
\begin{aligned}
P^{[\mathfrak{L}]}\big(\cancel{W}\big) &= \Pr\{|\ker(\mathsf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}| \geq 1\} && (3.22)\\
&= \sum_{j\geq 1} \Pr\{|\ker(\mathsf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}| = j\} && (3.23)\\
&\leq \sum_{j\geq 0} j \cdot \Pr\{|\ker(\mathsf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}| = j\}. && (3.24)
\end{aligned}
$$

Since the last line corresponds to the expected cardinality of the non-trivial kernel of G_{R}, the probability $P^{[\mathfrak{L}]}\big(\cancel{W}\big)$ can be upper bounded accordingly:

$$
P^{[\mathfrak{L}]}\big(\cancel{W}\big) \leq \mathrm{E}\{|\ker(\mathsf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}|\}. \tag{3.25}
$$

However, this bound can be tightened by a factor of $q-1$ by exploiting the fact that if some $\mathbf{x} \in \ker(\mathsf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}$, then also $a\mathbf{x} \in \ker(\mathsf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}$, $\forall a \in \mathbb{F}_q \setminus \{0\}$. So in order to bound (3.21) from above, it is sufficient to count just one of the $q-1$ scaled versions of $\mathbf{x}$

$$
\begin{aligned}
P^{[\mathfrak{L}]}\big(\cancel{W}\big) \leq \overline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big)&\\
&\triangleq \frac{1}{q-1} \cdot \mathrm{E}\{|\ker(\mathsf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}|\} && (3.26)
\end{aligned}
$$

and w.l.o.g. this is accomplished by counting only those vectors $\mathbf{x}$ that have been normalised w.r.t. their first non-zero entry, i.e. vectors $\mathbf{x}$ whose first non-zero entry is $x_i = 1$:

$$
\overline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big) = \sum_{\substack{\mathbf{x}\in\mathbb{F}_q^k,\\ x_i=1}} \Pr\{\mathsf{G}_{\mathrm{R}}\mathbf{x} = \mathbf{0}\}. \tag{3.27}
$$

The $k\gamma_{\mathrm{R}}$ rows of G_{R} can be viewed as the outcomes of independent trials of a random variable $\mathsf{r} \in \mathbb{F}_q^k$.

$$
\overline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big) = \sum_{\substack{\mathbf{x}\in\mathbb{F}_q^k,\\ x_i=1}} \Big[\Pr\big\{\mathsf{r}^{\mathsf{T}}\mathbf{x} = 0\big\}\Big]^{k\gamma_{\mathrm{R}}} \tag{3.28}
$$

The Hamming weight of a vector over $\mathbb{F}_q$ equals the number of non-zero elements and is denoted $\|\cdot\|_{\mathrm{H}}$. Now, the probability $\Pr\big\{\mathsf{r}^{\mathsf{T}}\mathbf{x} = 0\big\}$ is determined, conditioned on $\|\mathsf{r}\|_{\mathrm{H}} = d$ and $\|\mathbf{x}\|_{\mathrm{H}} = w$, where a row r has weight $\|\mathsf{r}\|_{\mathrm{H}} = d$ with probability Ω_d and there are $\binom{k}{w}(q-1)^{w-1}$ choices of $\mathbf{x}$ of weight $w > 0$ and a one as the first non-zero entry:

$$
\overline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big) = \sum_{w=1}^{k} \binom{k}{w} (q-1)^{w-1} \left[\sum_{d\in\mathcal{D}} \Omega_d \Pr\Big\{\mathsf{r}^{\mathsf{T}}\mathbf{x} = 0 \,\Big|\, \|\mathsf{r}\|_{\mathrm{H}} = d,\ \|\mathbf{x}\|_{\mathrm{H}} = w\Big\}\right]^{k\gamma_{\mathrm{R}}}. \tag{3.29}
$$

Let $\mathsf{v} = (\mathsf{v}_1, \mathsf{v}_2, \ldots, \mathsf{v}_k)^\mathsf{T}$ with $\mathsf{v}_j = \mathsf{r}_j x_j$, where v_j, r_j and x_j are the jth elements of the vectors v, r and $\mathbf{x}$, respectively, then

$$\Pr\left\{ \mathsf{r}^\mathsf{T}\mathbf{x} = 0 \,\middle|\, \|\mathsf{r}\|_\mathrm{H} = d,\ \|\mathbf{x}\|_\mathrm{H} = w \right\}$$
$$= \sum_{l=0}^{d} \Pr\left\{ \|\mathsf{v}\|_\mathrm{H} = l \,\middle|\, \|\mathsf{r}\|_\mathrm{H} = d,\ \|\mathbf{x}\|_\mathrm{H} = w \right\} \Pr\left\{ \sum_{j=1}^{k} \mathsf{v}_j = 0 \middle| \|\mathsf{v}\|_\mathrm{H} = l \right\}. \tag{3.30}$$

The probability of occurrence of exactly l non-zero elements in v is

$$\Pr\left\{ \|\mathsf{v}\|_\mathrm{H} = l \,\middle|\, \|\mathsf{r}\|_\mathrm{H} = d,\ \|\mathbf{x}\|_\mathrm{H} = w \right\} = \frac{\binom{w}{l}\binom{k-w}{d-l}}{\binom{k}{d}}. \tag{3.31}$$

The last term in (3.30) is the number $N_0(l, q)$ of possibilities that l non-zero $\mathbb{F}_q$-elements add up to zero, taking the elements' order into account, divided by the number $N(l, q)$ of all possibilities to draw l times with replacement from the set of the $q - 1$ non-zero $\mathbb{F}_q$-elements also taking the order into account:

$$\Pr\left\{ \sum_{j=1}^{k} \mathsf{v}_j = 0 \,\middle|\, \|\mathsf{v}\|_\mathrm{H} = l \right\} = \frac{N_0(l,\, q)}{N(l,\, q)}. \tag{3.32}$$

The problem of determining $N_0(l, q)$ is equivalent to finding the number of closed walks of length l in a complete graph of size q from some fixed but arbitrary vertex back to itself of which a closed form expression can be found, e.g. in [Sta11]

$$N_0(l,\, q) = \frac{1}{q}\left[(q-1)^l + (q-1)(-1)^l\right]. \tag{3.33}$$

With $N(l, q) = (q - 1)^l$ the expression in (3.32) results in

$$\Pr\left\{ \sum_{j=1}^{k} \mathsf{v}_j = 0 \,\middle|\, \|\mathsf{v}\|_\mathrm{H} = l \right\} = \frac{1}{q}\left[1 - (1-q)^{1-l}\right]. \tag{3.34}$$

Finally, inserting (3.31) and (3.34) into (3.30) and the resulting expression into (3.29) concludes the assertion. □

The second variant of the upper bound given by (3.19) is obtained by manipulating the term in (3.18) in the outer square brackets:

$$\begin{aligned}
&\frac{1}{q}\sum_{d\in\mathcal{D}}\Omega_d\cdot\frac{\sum\limits_{l=0}^{d}\binom{w}{l}\binom{k-w}{d-l}\left[1-(1-q)^{1-l}\right]}{\binom{k}{d}}\\
&=\frac{1}{q}\sum_{d\in\mathcal{D}}\Omega_d\cdot\frac{\sum\limits_{l=0}^{d}\binom{w}{l}\binom{k-w}{d-l}+(q-1)\sum\limits_{l=0}^{d}(-1)^l(q-1)^{-l}\binom{w}{l}\binom{k-w}{d-l}}{\binom{k}{d}} && (3.35)\\
&=\frac{1}{q}\sum_{d\in\mathcal{D}}\Omega_d\cdot\left[1+(q-1)\frac{\sum\limits_{l=0}^{d}(-1)^l(q-1)^{d-l}\binom{w}{l}\binom{k-w}{d-l}}{(q-1)^d\binom{k}{d}}\right] && (3.36)\\
&=\frac{1}{q}+\frac{q-1}{q}\sum_{d\in\mathcal{D}}\Omega_d\cdot\frac{\mathcal{K}_d(w;k)}{\mathcal{K}_d(0;k)} && (3.37)
\end{aligned}$$

In (3.35) the Chu-Vandermonde identity

$$\binom{k}{d}=\sum_{l=0}^{d}\binom{w}{l}\binom{k-w}{d-l} \tag{3.38}$$

is used and in (3.36) the Krawtchouk polynomials $\mathcal{K}_d(w;k)$ and $\mathcal{K}_d(0;k)$ can be identified in the numerator and the denominator, respectively.

3.3.2 An Upper Bound on the Symbol Erasure Probability

Due to the striking similarity between the upper bound on the symbol erasure probability $\overline{P}^{[\mathfrak{L}]}\left(\cancel{S}\right)$ and the upper bound on the word erasure probability $\overline{P}^{[\mathfrak{L}]}\left(\cancel{W}\right)$, only the differing part of the proof will be given in detail. The following upper bound [SLV11, Lup11, SGV13] on symbol level is a generalisation of an expression for binary codes from [RVF07] to higher order Galois fields.

Theorem 3.15. Given an LT code ensemble $\mathfrak{L}=\left(\mathbf{G}_{\mathrm{R}}\sim\Omega(\xi),\,\mathbb{F}_q^{n_{\mathrm{R}}\times k}\right)$, an upper bound on the symbol erasure probability $P^{[\mathfrak{L}]}\left(\cancel{S}\right)$ after ML decoding is

$$\overline{P}^{[\mathfrak{L}]}\left(\cancel{S}\right)=\sum_{w=1}^{k-1}\binom{k-1}{w-1}(q-1)^{w-1}\left[\frac{1}{q}+\frac{q-1}{q}\sum_{d\in\mathcal{D}}\Omega_d\cdot\frac{\mathcal{K}_d(w;k)}{\mathcal{K}_d(0;k)}\right]^{k\gamma_{\mathrm{R}}} \tag{3.39}$$

with the inverse reception code rate $\gamma_{\mathrm{R}}=1+\varepsilon_{\mathrm{R}}$.

Proof. The probability $P^{[\mathfrak{L}]}\left(\cancel{S}\right)$ is equal to the probability that the ith input symbol cannot be determined by ML decoding for an arbitrary $i\in\{1,2,\ldots,k\}$

$$P^{[\mathfrak{L}]}\left(\cancel{S}\right)=\Pr\left\{\exists\mathbf{x}\in\mathbb{F}_q^k\setminus\{\mathbf{0}\},\,x_i=a\,:\,\mathbf{G}_{\mathrm{R}}\mathbf{x}=\mathbf{0}\right\} \tag{3.40}$$

with arbitrary but fixed $a \in \mathbb{F}_q \backslash \{0\}$. The right-hand side of (3.40) is the probability of the ith column of matrix $\mathbf{G}_{\mathrm{R}}$ being linearly dependent on a non-empty set of columns. This can be upper bounded by the probability that any possible set of columns of $\mathbf{G}_{\mathrm{R}}$ is linearly dependent on column i

$$P^{[\mathfrak{L}]}\big(\not\$\big) \leq \overline{P}^{[\mathfrak{L}]}\big(\not\$\big) = \sum_{\substack{x \in \mathbb{F}_q^k, \\ x_i = a}} \Pr\{\mathbf{G}_{\mathrm{R}} x = \mathbf{0}\}. \tag{3.41}$$

The remainder of this proof is along the same lines as the proof of Theorem 3.14 with the only difference that in contrast to the previous derivation, there are $\binom{k-1}{w-1}(q-1)^{w-1}$ choices of $\mathbf{x}$ of weight $w > 0$ with $x_i = a$. □

3.3.3 A Lower Bound on the Symbol Erasure Probability

Lemma 3.16 (from [SGV13])**.** Given an LT code ensemble $\mathfrak{L} = \big(\mathbf{G}_{\mathrm{R}} \sim \Omega(\xi), \mathbb{F}_q^{n_{\mathrm{R}} \times k}\big)$, the probability that i particular (fixed but arbitrary) input nodes (IN) are not connected to any of the $k\gamma_{\mathrm{R}}$ independent output nodes (ON), i.e. the probability that i particular columns of $\mathbf{G}_{\mathrm{R}}$ are all-zero columns, is given by

$$\Pr\{\mathsf{i} = i \text{ particular IN not connected to any ON}\} = \left(\sum_{d \in \mathcal{D}} \Omega_d \frac{\binom{k-i}{d}}{\binom{k}{d}} \right)^{k\gamma_{\mathrm{R}}}. \tag{3.42}$$

Proof. The probability that i particular input nodes, with $0 \leq i \leq k$, are not connected to an output node of degree d is

$$\frac{\binom{k-i}{d}}{\binom{k}{d}}, \tag{3.43}$$

while the probability that i particular input nodes are not connected to an output node of arbitrary degree is

$$\sum_{d \in \mathcal{D}} \Omega_d \frac{\binom{k-i}{d}}{\binom{k}{d}}. \tag{3.44}$$

Finally, since there are $k\gamma_{\mathrm{R}}$ independent output nodes, the probability that i particular input nodes are not connected to any of them is given by (3.42). □

The latter proof is similar to that for the special case $i = 1$ from [Sho06]. Since (3.42) with $i = 1$ constitutes the tightest known lower bound on the symbol erasure probability, it is briefly summarised in the following without proof.

Theorem 3.17 (from [Sho06])**.** Given an LT code ensemble $\mathfrak{L} = \left(\mathbf{G}_{\mathrm{R}} \sim \Omega(\xi),\, \mathbb{F}_q^{n_{\mathrm{R}} \times k}\right)$, a lower bound on the symbol erasure probability $P^{[\mathfrak{L}]}\left(\cancel{S}\right)$ after ML decoding is

$$\begin{aligned}\underline{P}^{[\mathfrak{L}]}\left(\cancel{S}\right) &= \Pr\{\mathsf{i} = 1 \text{ particular IN not connected to any ON}\} \\ &= \left(1 - \frac{\bar{d}}{k}\right)^{k\gamma_{\mathrm{R}}},\end{aligned} \tag{3.45}$$

with the average row weight $\bar{d}$.

3.3.4 A Lower Bound on the Word Erasure Probability

Theorem 3.18 (from [SGV13])**.** Given an LT code ensemble $\mathfrak{L} = \left(\mathbf{G}_{\mathrm{R}} \sim \Omega(\xi),\, \mathbb{F}_q^{n_{\mathrm{R}} \times k}\right)$, a lower bound on the word erasure probability $P^{[\mathfrak{L}]}\left(\cancel{W}\right)$ after ML decoding is

$$\underline{P}^{[\mathfrak{L}]}\left(\cancel{W}\right) = \sum_{i=1}^{k} (-1)^{i+1} \binom{k}{i} \left(\sum_{d \in \mathcal{D}} \Omega_d \frac{\binom{k-i}{d}}{\binom{k}{d}} \right)^{k\gamma_{\mathrm{R}}}. \tag{3.46}$$

Proof. An information word cannot be reconstructed if at least one input node cannot be recovered. A lower bound on the word erasure probability $P^{[\mathfrak{L}]}\left(\cancel{W}\right)$ is therefore given by the probability that there exist input nodes that are not connected to any of the $k\gamma_{\mathrm{R}}$ independent output nodes

$$\underline{P}^{[\mathfrak{L}]}\left(\cancel{W}\right) = \Pr\{\exists \text{ IN not connected to any ON}\} \tag{3.47}$$

$$= \sum_{j=1}^{k} \Pr\{\text{exactly } \mathsf{j} = j \text{ IN not connected to any ON}\}. \tag{3.48}$$

However, the individual summands in (3.48) are not given explicitly. They appear only implicitly in a different representation of (3.42)

$$\begin{aligned}&\Pr\{\mathsf{i} = i \text{ particular IN not connected to any ON}\} \\ &= \frac{\sum_{j=i}^{k} \binom{j}{i} \Pr\{\text{exactly } \mathsf{j} = j \text{ IN not connected to any ON}\}}{\binom{k}{i}},\end{aligned} \tag{3.49}$$

where the numerator denotes the probability that i *arbitrary* input nodes are not connected to any output nodes. It results from the fact that given exactly j unconnected input nodes, there are $\binom{j}{i}$ possibilities to choose i particular unconnected input nodes. The whole expression is then normalised by $\binom{k}{i}$, the number of possibilities to choose i particular input nodes from the set of all k input nodes.

Below, the following identity which arises from the symmetry of the binomial coefficients

$$\sum_{\varsigma=1}^{\nu}(-1)^{\varsigma+1}\binom{\nu}{\varsigma}=1 \tag{3.50}$$

is used twice, as well as the fact that $\binom{\nu}{\varsigma}>0$ if $\nu,\varsigma\in\mathbb{N}_0$ and $0\leq\varsigma\leq\nu$, and that $\binom{\nu}{\varsigma}=0$ in all other cases.

Multiplying (3.49) by $(-1)^{i+1}\binom{k}{i}$ and summing over all i yields

$$\begin{aligned}
&\sum_{i=1}^{k}(-1)^{i+1}\binom{k}{i}\Pr\{\mathsf{i}=i\text{ particular IN not connected to any ON}\}\\
&=\sum_{i=1}^{k}(-1)^{i+1}\sum_{j=i}^{k}\binom{j}{i}\Pr\{\text{exactly }\mathsf{j}=j\text{ IN not connected to any ON}\} \qquad (3.51)\\
&=\sum_{i=1}^{k}\sum_{j=i}^{k}(-1)^{i+1}\binom{j}{i}\Pr\{\text{exactly }\mathsf{j}=j\text{ IN not connected to any ON}\}\\
&=\sum_{i=1}^{k}\sum_{j=1}^{k}(-1)^{i+1}\binom{j}{i}\Pr\{\text{exactly }\mathsf{j}=j\text{ IN not connected to any ON}\}\\
&=\sum_{j=1}^{k}\Pr\{\text{exactly }\mathsf{j}=j\text{ IN not connected to any ON}\}\sum_{i=1}^{k}(-1)^{i+1}\binom{j}{i}\\
&=\sum_{j=1}^{k}\Pr\{\text{exactly }\mathsf{j}=j\text{ IN not connected to any ON}\}\sum_{i=1}^{j}(-1)^{i+1}\binom{j}{i}\\
&=\sum_{j=1}^{k}\Pr\{\text{exactly }\mathsf{j}=j\text{ IN not connected to any ON}\}\\
&=\underline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big). \qquad (3.52)
\end{aligned}$$

Finally, inserting (3.42) into the left-hand side of (3.51) yields (3.46) and concludes the assertion. □

3.3.5 The Probability of Exactly j Unconnected Input Nodes

In (3.49) the probability $\Pr\{\text{exactly }\mathsf{j}=j\text{ IN not connected to any ON}\}$, which stands for the probability of exactly j input nodes not being connected to any of the $k\gamma_{\mathrm{R}}$ independent output nodes, is given only implicitly. Since it directly sums up to the lower bound on word level as in (3.48), one may call it a rather microscopic quantity in contrast to the macroscopic lower bound. As such it may be seen as an additional, more detailed benchmark for LT code ensembles. By applying some simple transformations, it is possible to obtain Pr{exactly $\mathsf{j} = j$ IN not connected to any ON} in explicit form from (3.49).

Lemma 3.19. Given an LT code ensemble $\mathfrak{L} = \left(\mathbf{G}_{\mathrm{R}} \sim \Omega(\xi),\, \mathbb{F}_q^{n_{\mathrm{R}} \times k}\right)$, the probability that exactly j input nodes are not connected to any of the $k\gamma_{\mathrm{R}}$ independent output nodes is

$$\begin{aligned} &\Pr\{\text{exactly } \mathsf{j} = j \text{ IN not connected to any ON}\} \\ &= \sum_{i=0}^{k} (-1)^{i+j} \binom{i}{j} \binom{k}{i} \left(\sum_{d \in \mathcal{D}} \Omega_d \frac{\binom{k-i}{d}}{\binom{k}{d}} \right)^{k\gamma_{\mathrm{R}}}. \end{aligned} \tag{3.53}$$

Proof. Rearranging (3.49) yields

$$\sum_{j=i}^{k} \binom{j}{i} P_{\mathrm{e}}(j) = \binom{k}{i} P_{\mathrm{p}}(i), \tag{3.54}$$

where the following short-hand notation has been used for an improved readability of the formulas:

$$\begin{aligned} P_{\mathrm{p}}(i) &\triangleq \Pr\{\mathsf{i} = i \text{ particular IN not connected to any ON}\} \\ P_{\mathrm{e}}(j) &\triangleq \Pr\{\text{exactly } \mathsf{j} = j \text{ IN not connected to any ON}\}. \end{aligned}$$

Multiplying (3.54) with $(-1)^i \binom{i}{l}$, where l is an arbitrary but fixed integer and $0 \le l \le i$, as well as summing over all i results in

$$\sum_{i=0}^{k} (-1)^i \binom{i}{l} \sum_{j=i}^{k} \binom{j}{i} P_{\mathrm{e}}(j) = \sum_{i=0}^{k} (-1)^i \binom{i}{l} \binom{k}{i} P_{\mathrm{p}}(i). \tag{3.55}$$

With $\sum\limits_{j=i}^{k} \binom{j}{i} P_{\mathrm{e}}(j) = \sum\limits_{j=0}^{k} \binom{j}{i} P_{\mathrm{e}}(j)$ and a further rearrangement, the following expression is obtained:

$$\sum_{j=0}^{k} P_{\mathrm{e}}(j) \sum_{i=0}^{k} (-1)^i \binom{i}{l} \binom{j}{i} = \sum_{i=0}^{k} (-1)^i \binom{i}{l} \binom{k}{i} P_{\mathrm{p}}(i). \tag{3.56}$$

For the inner summation on the left hand side the binomial identity

$$\sum_{\varsigma=0}^{\nu} (-1)^{\varsigma} \binom{\varsigma}{\xi} \binom{\nu}{\varsigma} = \begin{cases} 0 & \text{if } \xi < \nu \\ (-1)^{\nu} & \text{if } \xi = \nu \end{cases} \tag{3.57}$$

is used, which for instance can be found in [Gou72, eq. (3.119)] or [Gou10, eq. (10.9)]:

$$(-1)^l P_\mathrm{e}(l) = \sum_{i=0}^{k} (-1)^i \binom{i}{l} \binom{k}{i} P_\mathrm{p}(i). \tag{3.58}$$

Multiplying both sides with $(-1)^l$ and renaming l for j yields

$$P_\mathrm{e}(j) = \sum_{i=0}^{k} (-1)^{i+j} \binom{i}{j} \binom{k}{i} P_\mathrm{p}(i). \tag{3.59}$$

And finally, $P_\mathrm{p}(i)$ is replaced with the expression in (3.42) which results in (3.53).

□

3.4 Upper Bounds for the Random Ensembles

From the expressions in (3.19), (3.39), (3.45) and (3.46) it is straightforward to determine bounds for some of the special LT code ensembles by simply inserting the respective row weight distributions. The standard and the expurgated random ensembles are of particular interest. And although exact expressions for the word erasure probabilities of both random ensembles are given by (3.13) and (3.14), it is useful to derive upper bounds on their erasure probabilities, especially as the recursion in (3.14) is computationally complex even for quite small values of k.

Note that, as already depicted in Figure 3.1, (3.13) and (3.14) quickly converge to the same constant values as the input size k increases, with a given field size and a given number of received symbols. Moreover, the erasure probabilities of the expurgated ensemble are tightly upper bounded by those of the corresponding standard random ensemble, and thus upper bounds for the standard ensemble are also upper bounds for the expurgated ensemble. Therefore, the expurgated random ensemble will not be further emphasised in the following, but is accordingly implied when discussing the standard random ensemble. The following two corollaries are obtained by inserting (2.9) into (3.39) and (3.19), respectively:

Corollary 3.20. An upper bound on the symbol erasure probability $P^{[\mathfrak{L}]}\big(\cancel{S}\big)$ of the standard random ensemble under ML decoding is [SLV11]

$$\overline{P}^{[\mathfrak{L}]}\big(\cancel{S}\big) = q^{-k(\gamma_\mathrm{R}-1)-1} = q^{-\eta_\mathrm{R}-1}, \tag{3.60}$$

where $\eta_\mathrm{R} = k\varepsilon_\mathrm{R} = k(\gamma_\mathrm{R} - 1)$ is the absolute symbol reception overhead.

Corollary 3.21. An upper bound on the word erasure probability $P^{[\mathfrak{L}]}\big(\cancel{W}\big)$ of the standard random ensemble under ML decoding is [LPC10]

$$\overline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big) = \frac{1}{q-1} q^{-k(\gamma_\mathrm{R}-1)} = \frac{1}{q-1} q^{-\eta_\mathrm{R}}. \tag{3.61}$$

3.5 Numerical Evaluation and Monte Carlo Simulations

Having laid the theoretical basis for the finite length analysis of general LT code ensembles under optimal erasure decoding, first of all the just derived bounds on the residual erasure probability will be evaluated and contrasted with corresponding Monte Carlo simulations of residual erasure rates for different LT code ensembles. Then, it will be discussed, how to use these bounds for code design. And finally, the decoding complexity of some ensembles will be assessed.

3.5.1 The Standard Random Ensemble

It is an interesting feature of the two random ensembles, that their erasure correction performance is nearly independent of the input size k. It depends almost solely on the absolute number η_{R} of additionally received symbols and on the field size q. This property becomes evident when considering the respective right hand sides of (3.61) and (3.60). These upper bounds are depicted in Figure 3.2 for different field orders as functions of the absolute as well as the relative reception overhead, respectively.

In Figure 3.2(a) the upper bounds (3.61) and (3.60) are shown in terms of the absolute symbol reception overhead η_{R}. In this form, $\overline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big)$ and $\overline{P}^{[\mathfrak{L}]}\big(\cancel{S}\big)$ are independent of the input size k. Though, one has to be careful with drawing the conclusion that by using higher order Galois fields, lower erasure probabilities could be reached much faster [LPC10, QUA10]. In Figure 3.2(a) it is not taken into account that the reception of η_{R} additional symbols is not comparable for ensembles over different fields, since symbols consist of different numbers of bits.

To account for this discrepancy, in this thesis overhead-failure plots are preferably depicted in terms of the relative reception overhead ε_{R} or equivalently the inverse reception code rate $\gamma_{\mathrm{R}} = 1 + \varepsilon_{\mathrm{R}}$. In the case of the exemplary standard random ensembles over $\mathbb{F}_2$ to $\mathbb{F}_{256}$ with input sizes[2] $k = \frac{840}{\mathrm{ld}(q)} = \frac{840}{\mu} \in \{840,\, 420, \ldots,\, 105\}$, the upper bounds (3.61) and (3.60) are provided in Figure 3.2(b) as functions of ε_{R} and γ_{R}. These ensembles have in common that the information words contain the same number k_{B} of *bits* which is a crucial constraint to enable a fair comparison (cf. Section 2.1.1).

The Tightness of the Upper Bounds

In addition to the upper bounds on word level, the exact word erasure probabilities of the respective ensembles are included in Figure 3.2(c) in order to visualise how

[2] The input size of 840 bits has merely been chosen, since it is the least common multiple of $\mu \in \{1,\, 2, \ldots,\, 8\}$. This way ensembles over different fields have exactly the same number k_{B} of input bits. Using arbitrary values for k_{B}, the input sizes in terms of $\mathbb{F}_q$-elements then become $k = \lceil \frac{k_{\mathrm{B}}}{\mathrm{ld}(q)} \rceil = \lceil \frac{k_{\mathrm{B}}}{\mu} \rceil$.

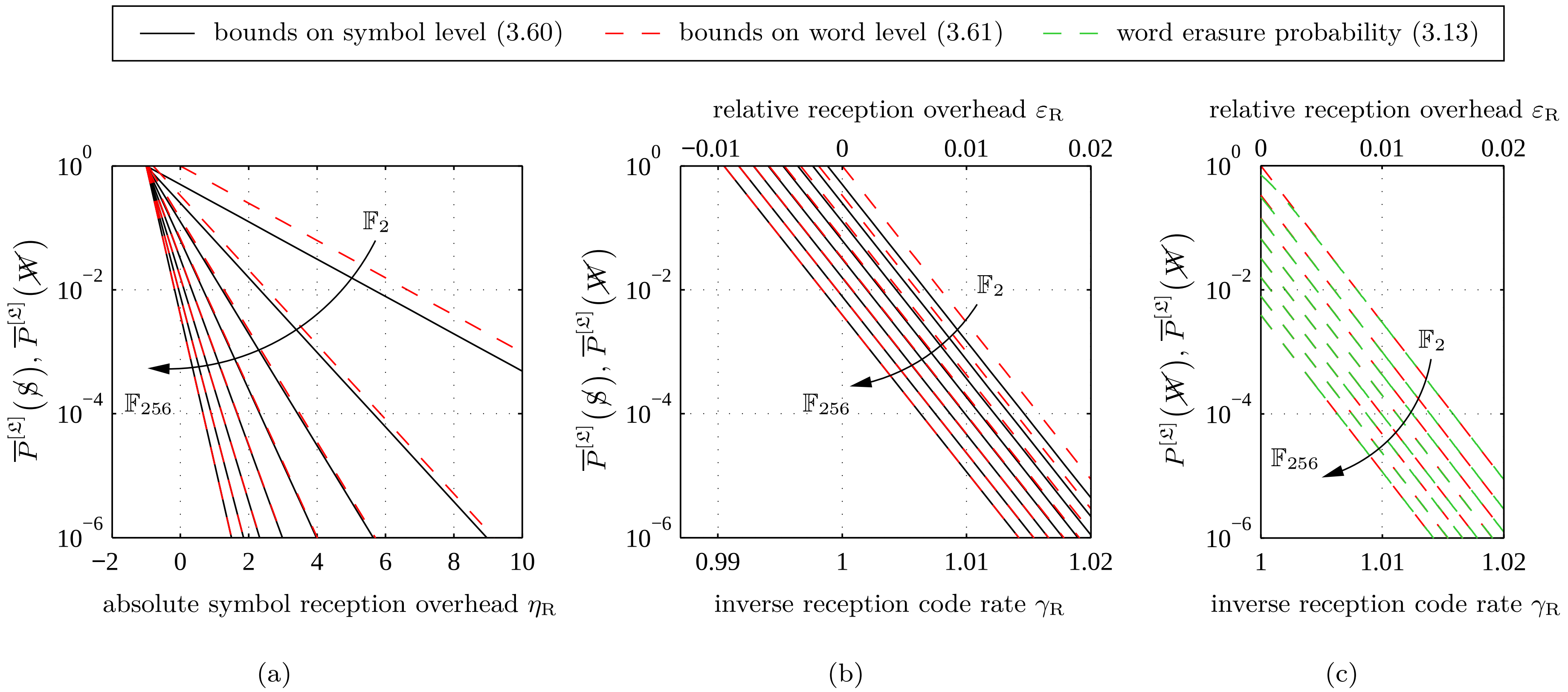


Figure 3.2: Upper bounds on symbol as well as on word level of the standard random ensemble over $\mathbb{F}_{2^\mu}$ with $\mu \in \{1, 2, \dots, 8\}$.

(a) Upper bounds as a function of the absolute overhead $\eta_{\mathrm{R}} = k\varepsilon_{\mathrm{R}}$.

(b) Upper bounds as a function of the inverse code rate γ_{R} or the relative overhead ε_{R}, respectively, for input sizes $k = \frac{840}{\mathrm{ld}(q)} = \frac{840}{\mu} \in \{840, 420, \dots, 105\}$.

(c) Upper bounds on word level as well as the word erasure probability depicted as a function of γ_{R} or ε_{R}, respectively, for input sizes $k = \frac{840}{\mathrm{ld}(q)} = \frac{840}{\mu} \in \{840, 420, \dots, 105\}$ in order to illustrate the tightness of the upper bounds. The upper bounds on symbol level have the same quality and are therefore not depicted here.

close these bounds are to the exact values. The upper bounds on symbol level, similarly close to the respective exact symbol erasure rates, are not displayed in the box. Therefore, mostly these upper bounds will be used hereafter instead of the exact probabilities. For the standard random ensemble the lower bounds are of no real practical use, since this ensemble does not show an erasure floor. In fact, as can be seen for instance in Figure 3.2, the erasure probability of this ensemble is virtually log-linear in the reception overhead.

The Expected Absolute Reception Overhead

Besides the residual erasure probability, the expected absolute *symbol* reception overhead $\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\}$ for successful decoding is an important characteristic number for an ensemble $\mathfrak{L}$ as well as the expected absolute *bit* reception overhead $\mu\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\}$. The expected absolute *symbol* reception overhead is given by

$$\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\} = \sum_{\eta_\mathrm{R}=0}^{\infty} \eta_\mathrm{R} \Pr\{\text{decoding success on word level exactly at } \eta_\mathrm{R}\}. \tag{3.62}$$

To be more specific, successful decoding exactly at η_R implies that decoding fails up to overhead $\eta_\mathrm{R} - 1$. Now, the probability of successful decoding exactly at η_R can be expressed by means of the decoding success probability $P^{[\mathfrak{L}]}(\,\mathrm{W}\,) = 1 - P^{[\mathfrak{L}]}\big(\cancel{\mathrm{W}}\big)$ on word level, which denotes the probability that an information word can be recovered after the reception of $k + \eta_\mathrm{R}$ encoded symbols. Note that this includes that the considered information word might be recoverable with less than $k + \eta_\mathrm{R}$ received symbols. Both the decoding failure probability $P^{[\mathfrak{L}]}\big(\cancel{\mathrm{W}}\big)$ as well as the decoding success probability $P^{[\mathfrak{L}]}(\,\mathrm{W}\,)$ are inherently parametrised with the overhead $\eta_\mathrm{R} = n_\mathrm{R} - k$. Since this dependence is required to be explicit here, they shall be denoted $P^{[\mathfrak{L}]}\big(\cancel{\mathrm{W}}\,;\eta_\mathrm{R}\big)$ and $P^{[\mathfrak{L}]}(\,\mathrm{W}\,;\eta_\mathrm{R})$ for the current consideration. So the expected absolute *symbol* reception overhead can be written as

$$\begin{aligned}
\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\} &= \sum_{\eta_\mathrm{R}=1}^{\infty} \eta_\mathrm{R} \left[P^{[\mathfrak{L}]}(\,\mathrm{W}\,;\eta_\mathrm{R}) - P^{[\mathfrak{L}]}(\,\mathrm{W}\,;\eta_\mathrm{R}-1)\right] \\
&= \sum_{\eta_\mathrm{R}=1}^{\infty} \eta_\mathrm{R} \left[P^{[\mathfrak{L}]}\big(\cancel{\mathrm{W}}\,;\eta_\mathrm{R}-1\big) - P^{[\mathfrak{L}]}\big(\cancel{\mathrm{W}}\,;\eta_\mathrm{R}\big)\right] \\
&= \sum_{\eta_\mathrm{R}=0}^{\infty} P^{[\mathfrak{L}]}\big(\cancel{\mathrm{W}}\,;\eta_\mathrm{R}\big).
\end{aligned} \tag{3.63}$$

For the standard random ensemble the respective word erasure probability function $P^{[\mathfrak{L}]}\big(\cancel{\mathrm{W}}\,;\eta_\mathrm{R}\big)$ is given by (3.13) and with increasing input sizes k the expected reception overhead quickly converges to the values given in Table 3.1.

In the third column the upper bound on the expected symbol reception overhead which is derived analogously to (3.63) by using (3.61) instead of (3.13) is also tabularised. The respective upper bound on the overhead in terms of bits is provided in the fifth column. The upper bounds on the residual erasure probability are extremely close to the exact values. This property is inherited to the expected overhead, which is visualised in Figure 3.7(a), where the exact as well as the respective upper bound on the expected absolute reception overhead $\mathrm{E}^{[\mathfrak{L}]}\{\eta_{\mathrm{R}}\}$ are depicted, both in terms of bits as well as $\mathbb{F}_q$-symbols.

3.5.2 The Sparse Random Ensembles

For a practically relevant parametrisation of the sparse random ensemble it is important to differentiate between the standard sparse and the expurgated sparse random ensemble. This is unlike the standard random ensemble, where the difference to the expurgated random ensemble is negligible.

The literature is usually focused on the theory of the standard sparse ensemble (cf. e.g. [Kol94, BKW97, AS08, Kol09]), since it arises from a very simple mathematical experiment, without constraints on the row (or column) weights. Due to the i.i.d. element-wise construction of the matrix, one can exploit the independence of both rows and columns, which yields some appealing closed form results. Nevertheless, the expurgation of all-zero rows from a standard sparse LT code matrix is definitely required, since for a coding application such rows are useless and occur with a

Table 3.1: Expected absolute symbol and bit reception overheads for the standard random ensemble (exact values and upper bounds) for not too small input sizes k ($k \gtrsim 10$).

	expected absolute symbol reception overhead $\mathrm{E}^{[\mathfrak{L}]}\{\eta_{\mathrm{R}}\}$		expected absolute bit reception overhead $\mu\mathrm{E}^{[\mathfrak{L}]}\{\eta_{\mathrm{R}}\}$	
Galois field	exact	upper bound	exact	upper bound
$\mathbb{F}_2$	1.606695152	2.0	1.606695152	2.0
$\mathbb{F}_4$	0.421097686	0.444444444	0.842195372	0.888888889
$\mathbb{F}_8$	0.160966184	0.163265306	0.482898552	0.489795918
$\mathbb{F}_{16}$	0.070848712	0.071111111	0.283394848	0.284444444
$\mathbb{F}_{32}$	0.033267085	0.033298647	0.166335425	0.166493235
$\mathbb{F}_{64}$	0.016121091	0.016124969	0.096726546	0.096749814
$\mathbb{F}_{128}$	0.007935535	0.007936016	0.055548745	0.055552112
$\mathbb{F}_{256}$	0.003936887	0.003936947	0.031495096	0.031495576

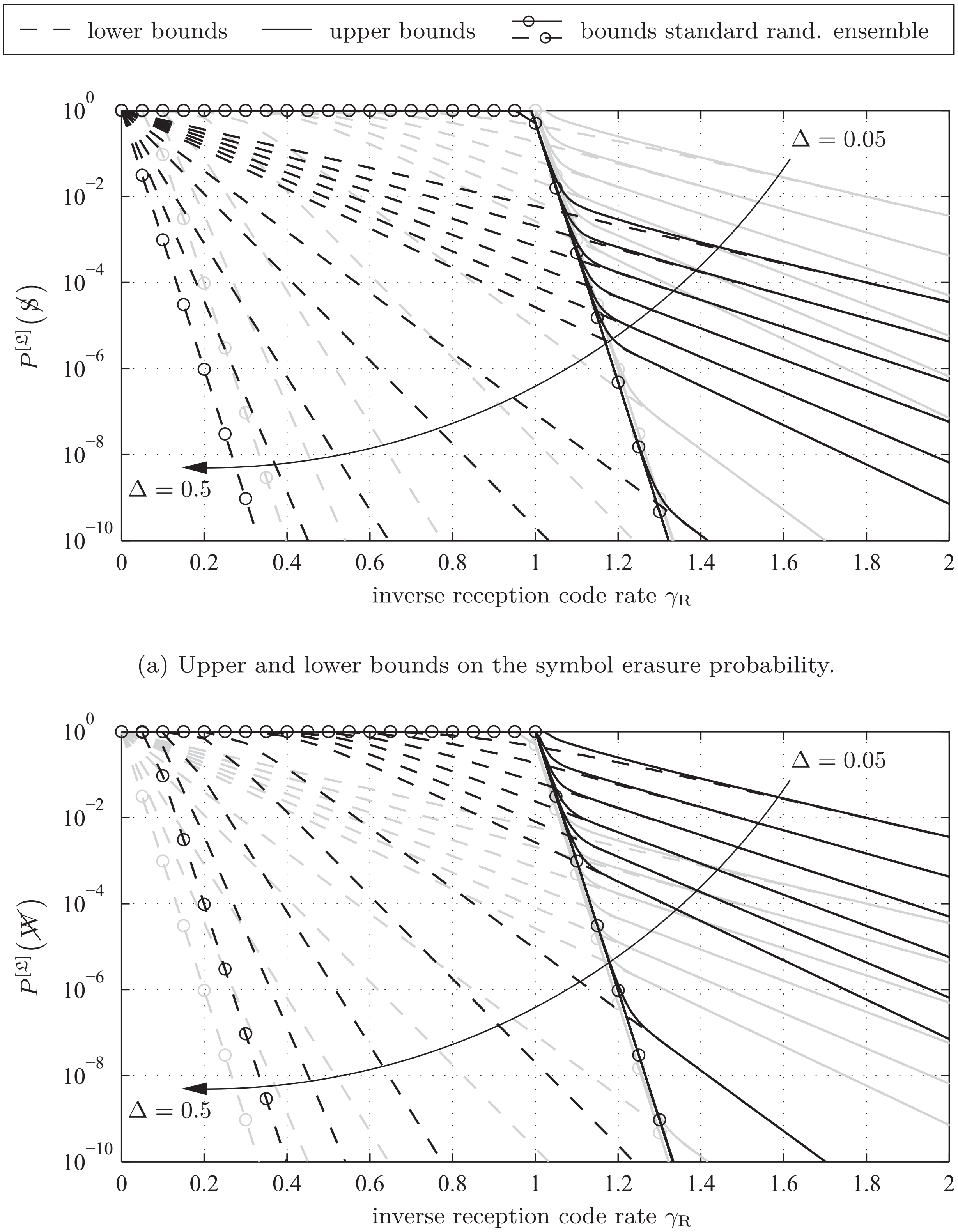
non-negligible probability. Therefore, when speaking about the sparse random ensemble in the following, the expurgated ensemble is meant.

In Figure 3.3 the upper and lower bounds on the erasure probability are depicted for binary sparse random ensembles of size $k = 100$. Like for the standard random ensemble, the upper bounds are so close to the true erasure probabilities, which will be shown subsequently in this section, that they will be used in lieu thereof. Note that here the erasure probabilities are not given analytically, but have to be determined by carrying out Monte Carlo simulations, i.e. by solving a huge number of systems of random linear equations by means of ML decoding. Since the number of simulated systems is chosen sufficiently high, the residual erasure rates can be taken for the true erasure probabilities in the provided figures.

In Figure 3.3(a) the bounds on symbol level are drawn in black and the bounds on word level are outlined in grey. In Figure 3.3(b) it is vice versa for a better comparison. The provided bounds are for ensembles with different densities $\Delta \in \{0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5\}$. The last value corresponds to the standard random ensemble, i.e. the optimal ensemble under ML decoding. Its log-linear bounds are marked with circles in the two figures.

The erasure probabilities of the sparse random ensembles have a very characteristic form, i.e. the two regions waterfall and erasure floor are very distinct, i.e. the two regions are almost log-linear. In the waterfall region the erasure probability of the sparse random ensembles is essentially equal to that of the corresponding standard random ensemble, i.e. it is near-optimal in the waterfall in a general sense. In the erasure floor region the symbol (word) erasure probability is almost equal to the respective lower bound which depends (almost) only on the density Δ, and therefore has a near-optimal erasure correction performance in this region for a given density. Most other ensembles of the same density converge slower to the lower bounds, which on symbol level are equal for all ensembles of the same density and the lower bounds on word level, though not exactly equal, are almost indistinguishably similar.

The just mentioned approximate log-linearity in the two regions of the sparse random ensembles makes it extremely simple to design them for specific requirements, particularly on symbol level. Since the input size is usually given due to application constraints, the waterfall region is fixed and can be approximated by the log-line of the standard random ensemble. Thus, the only remaining free parameter is the density or the average row weight and eventually the field size. So by specifying the maximum allowed erasure probability that should be reached at the end of the waterfall, a point on the log-line of the standard random ensemble is defined. Now it remains to find the lower bound log-line that passes through this point. The density that leads to that lower bound is the wanted density of the sparsest random ensemble that (almost) fulfils the requirements. Since the transition from waterfall to erasure floor region is continuous and leaves a small gap (which de-



(a) Upper and lower bounds on the symbol erasure probability.

(b) Upper and lower bounds on the word erasure probability.

Figure 3.3: Upper and lower bounds on the erasure probability on symbol and on word level for binary (sparse) random ensembles of size $k = 100$ with the following densities: $\Delta \in \{0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5\}$.

pends on the density) to the two characteristic log-lines, a slightly higher density might be necessary.

All binary LT code ensembles with a density lower than 0.5 show an erasure floor, though the erasure floors of the denser ensembles start only below the depicted erasure probability of 10^{-10}. In fact, binary ensembles with a higher density than 0.5 also show an erasure floor. In general, for non-binary codes this critical density is $\Delta = 1 - \frac{1}{q}$, i.e. the density of the respective standard random ensemble. Beyond this density, the probability to obtain full column rank matrices decreases again.

The Tightness of the Upper Bounds

It has already been mentioned that the upper bounds of the sparse random ensemble, like the ones of the standard ensemble, are so close to the true values that they are actually useful for code design. To support this statement, simulated residual erasure rates as well as upper and lower bounds on word and on symbol level are depicted in Figure 3.4 for a binary ensemble and a non-binary ensemble over $\mathbb{F}_{64}$ as a function of the inverse reception code rate γ_{R}. The input sizes are 300 bits and 50 symbols, respectively. The average row weight is $\bar{d} = 10$ in both cases.

Due to the discrete nature of the reception process, the depicted simulated residual erasure rates are piecewise linearly connected for a better visualisation. This becomes particularly obvious for higher order fields. The linear pieces connect the points with overheads that correspond to an integer number of received output symbols. The calculation of the bounds has been limited to such values, too, although they can be evaluated for an arbitrary real-valued reception overhead.

The simulated values are highlighted with red plus markers, while the red lines connecting the markers are almost completely covered by the upper bounds in black. The only visible part is around $\gamma_{\mathrm{R}} \approx 1$. Already for γ_{R} slightly greater than one, the difference to the upper bound becomes invisible and negligible. So, besides the fast convergence of upper and lower bounds in the erasure floor region, this strong congruence between the upper bounds and the respective true values allows to use the upper bounds for code design and analysis instead of performing time-consuming Monte Carlo simulations.

Code Ensemble Sets

Sparse random ensembles have some very specific properties that need to be taken into account when designing or evaluating them for an application. From Figure 3.4 it can be observed that the waterfalls of codes over different fields have the same slope, both on symbol and on word level, if the condition is met that the number of input *bits* k_{B} is kept constant. Below, the latter condition is always assumed

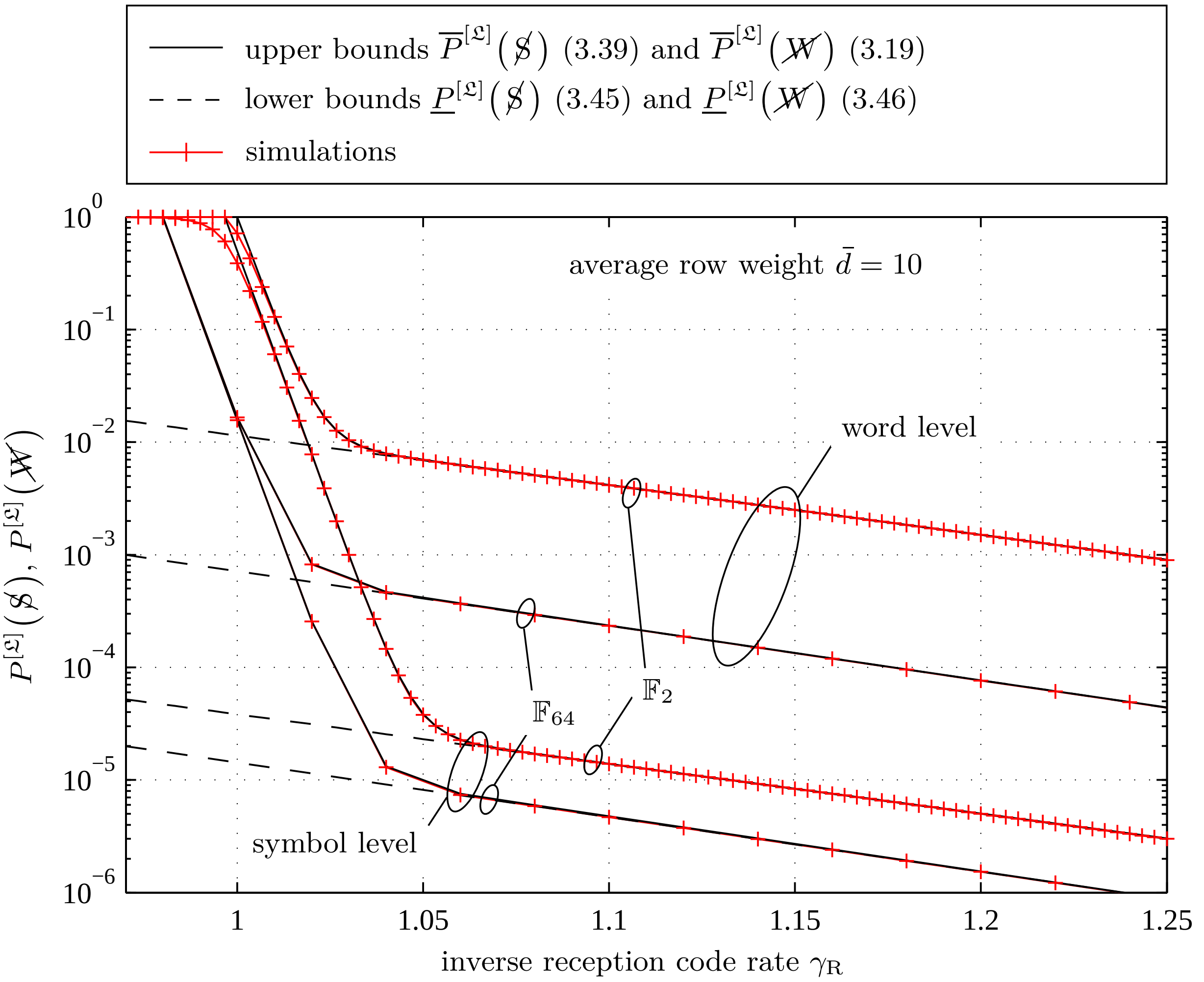


Figure 3.4: Bounds for the sparse random ensemble over $\mathbb{F}_2$ and $\mathbb{F}_{64}$ with $k = 300$ and $k = 50$ symbols, respectively, for $\bar{d} = 10$ together with the corresponding simulated residual erasure rates on symbol and on word level. The discrete nature of the output nodes becomes visible in the piecewise linear characteristic of the upper bounds and the simulated results. Additionally, due to the small deviation of the simulated erasure rates from their respective upper bounds, it is justified to describe the performance of sparse random ensembles by means of their upper bounds.

to be met. But apart from the waterfall, the code ensembles behave differently, depending on some further constraints. In the following, code ensembles with different constraints will be analysed and compared with each other. Ensembles over different fields that fulfil certain constraints will be subsumed to ensemble sets. For instance the two ensembles from Figure 3.4 are two elements from the ensemble set with equal average row weight $\bar{d} = 10$.

In the latter ensemble sets, codes over higher fields outperform their binary counterparts on symbol level, but even more on word level. Nevertheless, at least the respective symbol erasure probabilities are approximately in the same order of magnitude. The bounds on the symbol erasure probabilities of two exemplary

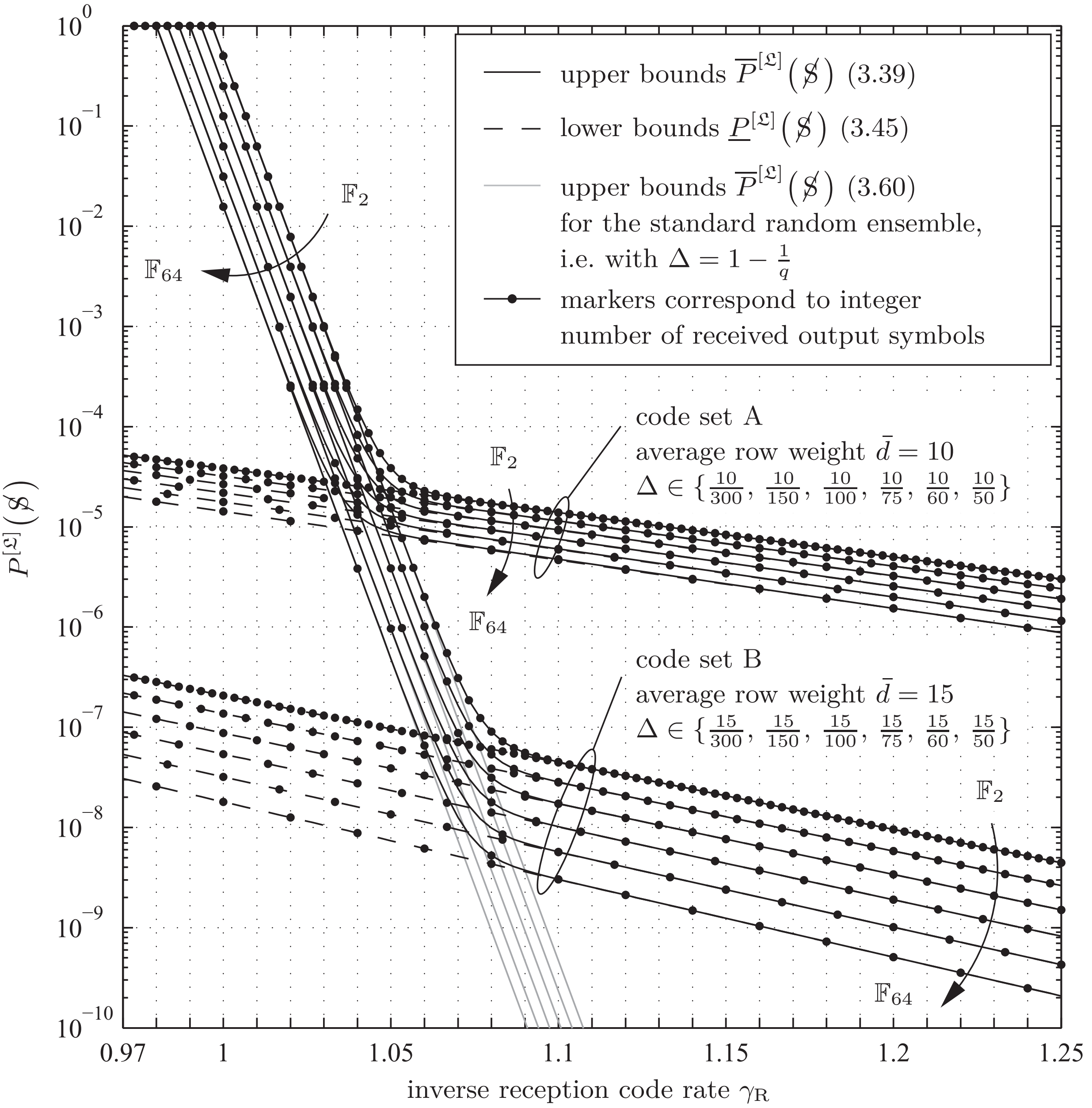


Figure 3.5: Upper and lower bounds on the symbol erasure probabilities for sparse random ensembles with input sizes $k = 300/\mathrm{ld}(q) = 300/\mu \in \{300, 150, 100, 75, 60, 50\}$ over Galois fields $\mathbb{F}_2$, $\mathbb{F}_4$, $\mathbb{F}_8$, $\mathbb{F}_{16}$, $\mathbb{F}_{32}$ and $\mathbb{F}_{64}$ as well as with average row weights $\bar{d} = 10$ and $\bar{d} = 15$, respectively.

ensemble sets with $\bar{d} = 10$ (ensemble set A) and $\bar{d} = 15$ (ensemble set B) up to $\mathbb{F}_{64}$ are depicted in Figure 3.5. A general lowering of the erasure floors with an increasing average row weight can be observed, but also within an ensemble set the erasure floor lowers as the field order increases. While the erasure floor spread within ensemble set A is relatively small, it becomes more relevant in ensemble set B, i.e. for higher average row weights.

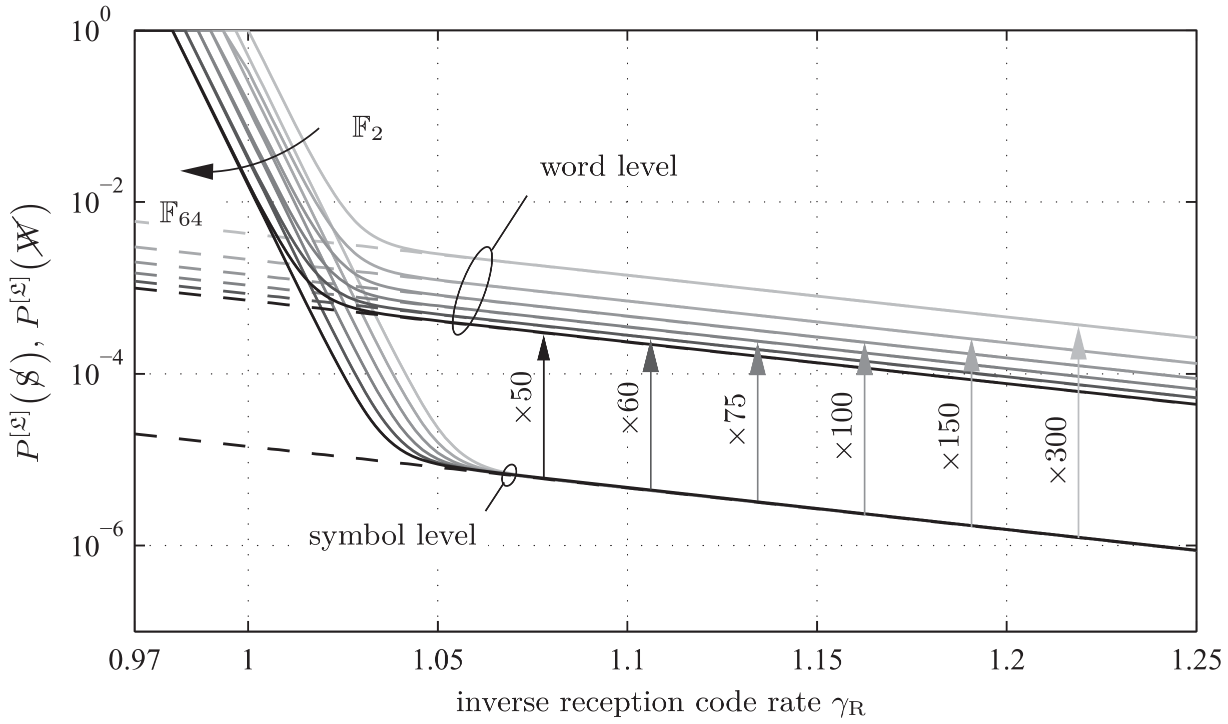
Practically it is more important to be able to choose from an ensemble set with equal erasure correction properties. However, completely equal erasure correction properties over different fields cannot be achieved, since the waterfalls are usually given by the constraints on the input size. On symbol level the waterfalls are log-parallel and log-equidistant for two successive field orders, while on word level they are only log-parallel, but converge to log-equidistance with increasing field size. Yet, it is possible to equalise the erasure floors either on symbol level or on word level by specifically tuning the density or the average row weights. These ensemble sets, whose erasure correction characteristics can be found in Figure 3.6, are thus either defined by the constraint of equal erasure floors on symbol level or on word level. Note that the two constraints cannot be fulfilled at the same time.

The ensembles in the ensemble set depicted in Figure 3.6(a) have the same erasure floor on symbol level. As a reference for the erasure floor, the 64-ary code from Figure 3.4 is used, i.e. the sparse random ensemble over $\mathbb{F}_{64}$ with 50 input symbols and average row weight $\bar{d} = 10$. The densities of the other ensembles from this set are obtained from equating the respective lower bound (3.45) with the one of the reference ensemble. The quantities related to the dependent ensemble are marked with a star:

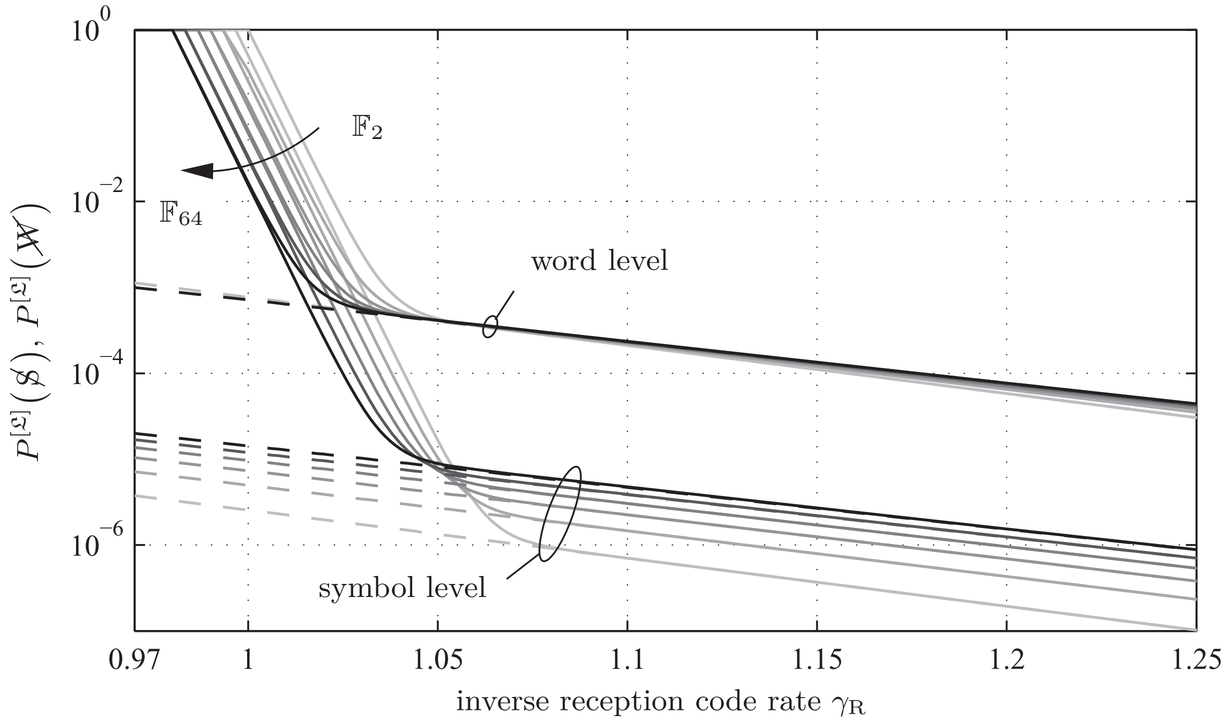
$$\begin{aligned} & \left(1 - \frac{\bar{d}^\star}{k^\star}\right)^{k^\star \gamma_\mathrm{R}} \overset{!}{=} \left(1 - \frac{\bar{d}}{k}\right)^{k\gamma_\mathrm{R}} \\ \Rightarrow \quad & (1 - \Delta^\star)^{k^\star \gamma_\mathrm{R}} = (1 - \Delta)^{k\gamma_\mathrm{R}} \\ \Rightarrow \quad & \Delta^\star = 1 - (1 - \Delta)^{\frac{k}{k^\star}} . \end{aligned} \tag{3.64}$$

In Figure 3.6(a) a particular relation between the respective erasure floors on symbol and on word level becomes evident: apart from being almost log-parallel, the distance is approximately equal to the respective input size. This fact reveals that the erasure floor is dominated by single residual symbol erasures. Note that single residual erasures do not occur due to linear dependencies between different columns, but because exactly one input symbol does not participate in any of the linear equations, i.e. there exists exactly one all-zero column in the decoding matrix. So one can expect that already by precoding with a single parity check (SPC) code, which is known to guarantee the correction of exactly one erasure, it should be possible to lower the erasure floors significantly at a very small rate loss of $(k - 1)/k$, when k is the input size of the LT code. In this chapter, this brief remark on precoding shall suffice as this topic will be further detailed in Chapter 6.

Constructing ensemble sets with equal erasure floors on word level is slightly more involved, since the lower bound on word level (3.46) does not exhibit a simple dependence on $\bar{d}$ or on Δ as does the lower bound on symbol level. To find such a



(a) Ensemble set with equal symbol erasure floor.



(b) Ensemble set with equal word erasure floor at $\gamma_R \approx 1.04$.

Figure 3.6: Upper and lower bounds for two ensemble sets with equal erasure floors. The reference code in both sets is the 64-ary sparse random ensemble with $k = 50$ and $\bar{d} = 10$. The respective input sizes are $k = 300/\mathrm{ld}(q) = 300/\mu \in \{300, 150, 100, 75, 60, 50\}$.

dependence, it helps to expand (3.46)

$$\begin{aligned}
\underline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big) &= \sum_{i=1}^{k}(-1)^{i+1}\binom{k}{i}\left(\sum_{d\in\mathcal{D}}\Omega_d\frac{\binom{k-i}{d}}{\binom{k}{d}}\right)^{k\gamma_{\mathrm{R}}} \\
&= k\left(\sum_{d\in\mathcal{D}}\Omega_d\frac{\binom{k-1}{d}}{\binom{k}{d}}\right)^{k\gamma_{\mathrm{R}}} - \binom{k}{2}\left(\sum_{d\in\mathcal{D}}\Omega_d\frac{\binom{k-2}{d}}{\binom{k}{d}}\right)^{k\gamma_{\mathrm{R}}} \pm \ldots \\
&\quad + (-1)^k k\left(\sum_{d\in\mathcal{D}}\Omega_d\frac{\binom{1}{d}}{\binom{k}{d}}\right)^{k\gamma_{\mathrm{R}}} + (-1)^{k+1}\left(\sum_{d\in\mathcal{D}}\Omega_d\frac{\binom{0}{d}}{\binom{k}{d}}\right)^{k\gamma_{\mathrm{R}}}.
\end{aligned}$$

Evaluating the individual terms for practically relevant parameter sets (and here the term practically relevant can be understood in a very broad sense) it shows that the first order term is already an extremely good approximation

$$\underline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big) \approx k\left(\sum_{d\in\mathcal{D}}\Omega_d\frac{\binom{k-1}{d}}{\binom{k}{d}}\right)^{k\gamma_{\mathrm{R}}}. \tag{3.65}$$

This approximation now contains the lower bound on symbol level, i.e.

$$\underline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big) \approx k\underline{P}^{[\mathfrak{L}]}\big(\cancel{S}\big) = k\left(1-\frac{\bar{d}}{k}\right)^{k\gamma_{\mathrm{R}}}. \tag{3.66}$$

So in order to simplify the construction it has proven sufficiently accurate to use this first order approximation of $\underline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big)$. By equating the approximation of $\underline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big)$ of the reference and the dependent ensembles, with the quantities of the latter ones, again marked with a star, the appropriate densities can be obtained.

$$\begin{aligned}
& k^\star\left(1-\frac{\bar{d}^\star}{k^\star}\right)^{k^\star\gamma_{\mathrm{R}}} = k\left(1-\frac{\bar{d}}{k}\right)^{k\gamma_{\mathrm{R}}} \\
\Rightarrow\quad & k^\star(1-\Delta^\star)^{k^\star\gamma_{\mathrm{R}}} = k(1-\Delta)^{k\gamma_{\mathrm{R}}} \\
\Rightarrow\quad & (1-\Delta^\star)^{k^\star\gamma_{\mathrm{R}}} = \frac{k}{k^\star}(1-\Delta)^{k\gamma_{\mathrm{R}}} \\
\Rightarrow\quad & \Delta^\star = 1-\left(\frac{k}{k^\star}\right)^{\frac{1}{k^\star\gamma_{\mathrm{R}}}}(1-\Delta)^{\frac{k}{k^\star}}
\end{aligned} \tag{3.67}$$

Note that equality can only be achieved for one particular overhead at the same time. In Figure 3.6(b) equality of the lower bounds is given at $\gamma_{\mathrm{R}} = 1.04$. For higher values of γ_{R} the erasure probabilities diverge slowly.

The Expected Absolute Symbol Reception Overhead

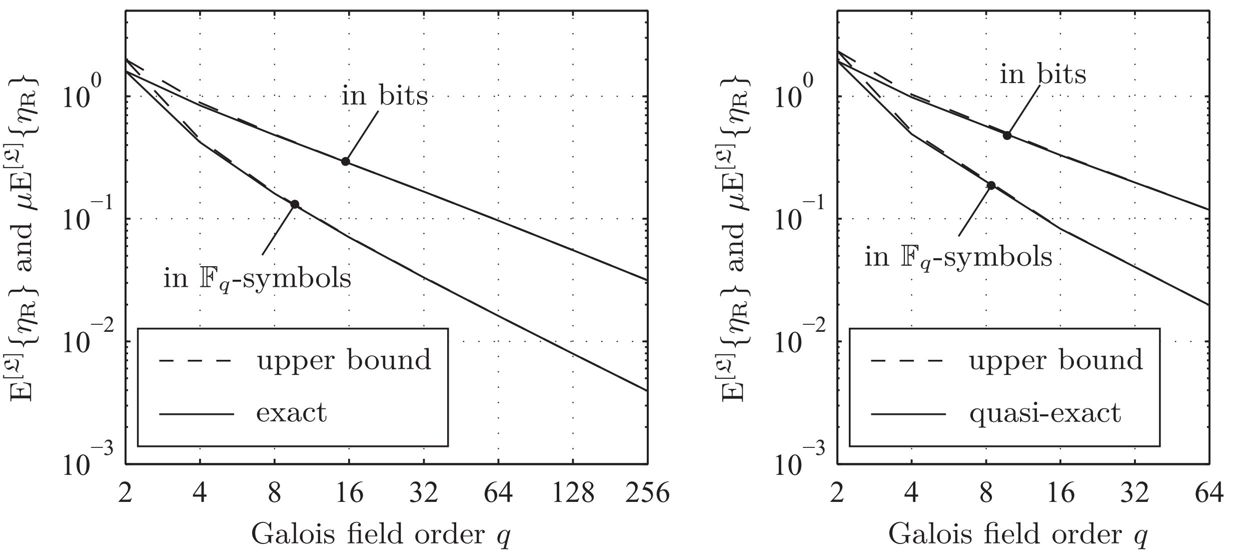
The exact word erasure probability is not known analytically for other LT ensembles than the standard random ensembles, so it has to be determined using simulations. However, as the erasure floors can be quite shallow and as the number of relevant terms in (3.63) can then be very large, the determination of the expected absolute symbol reception overhead $\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\}$ may lead to very long simulation times.

A quicker method for the sparse random ensemble would be to use again the upper bounds on word level instead of the exact word erasure probabilities. Although this already yields quite good results particularly for non-binary ensembles, it can be improved significantly. As the upper bound has the largest distance (both absolute as well as relative) to the true value at zero overhead and this distance decreases quickly and strictly monotonic with the overhead, it is advisable to perform a few simulations to determine $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}};\eta_\mathrm{R}\big)$ for the first couple of overhead values and for the remaining η_R one can use the respective upper bound.

In Table 3.2 $\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\}$ and $\mu\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\}$ has been calculated for ensemble set A, i.e. the sparse random ensemble set with $\bar{d} = 10$ and $k = 300/\mu$ input symbols. In the second and the fourth column, the combination of simulation and upper bound, as described above, has been used. Those values are almost equal to the true values and are thus treated as quasi-exact in the following. The values in columns three

Table 3.2: Expected absolute symbol and bit reception overheads for the ensemble set A (cf. Figure 3.5). The second and fourth column can be assumed to contain the true values. They are calculated by using simulated values for small values of η_R and continuing with the upper bounds as soon as $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}};\eta_\mathrm{R}\big)$ and $\overline{P}^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}};\eta_\mathrm{R}\big)$ have converged sufficiently. For the third and fifth column only the upper bounds are used.

Galois field	expected absolute symbol reception overhead $\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\}$		expected absolute bit reception overhead $\mu\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\}$	
	quasi-exact	upper bound	quasi-exact	upper bound
$\mathbb{F}_2$	1.932958	2.334474	1.932958	2.334474
$\mathbb{F}_4$	0.490783	0.516869	0.981566	1.033738
$\mathbb{F}_8$	0.187236	0.189988	0.561708	0.569964
$\mathbb{F}_{16}$	0.082891	0.083480	0.331564	0.333920
$\mathbb{F}_{32}$	0.039545	0.039751	0.197725	0.198755
$\mathbb{F}_{64}$	0.019674	0.019740	0.118044	0.118440



(a) Standard random ensemble with not too small values of k, i.e. $k \geq 10$.

(b) Sparse random ensemble set with $\bar{d} = 10$ and 300 input bits.

Figure 3.7: Expected value of the absolute reception overhead (3.62) for the standard random ensembles and the sparse random ensembles from the ensemble set A, i.e. with $\bar{d} = 10$ and $k = 300/\mu$ input symbols. The expected values are given in terms of $\mathbb{F}_q$-symbols and in bits.

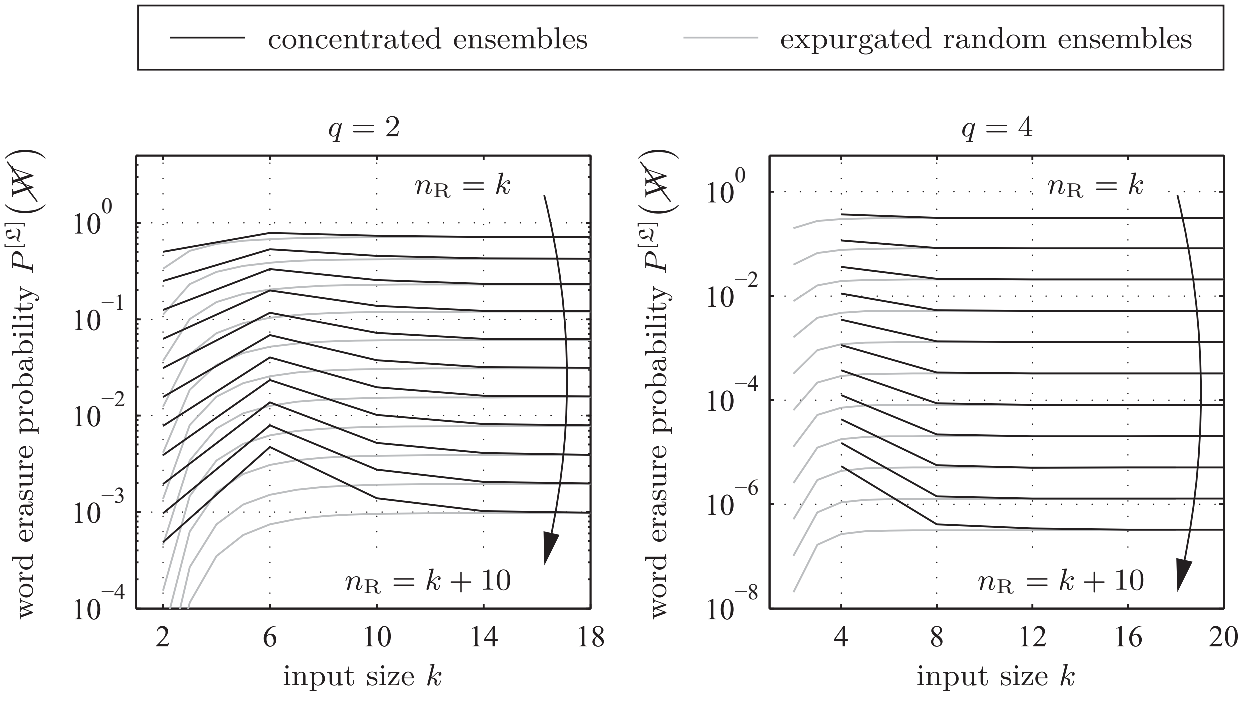
and five are obtained from using $\overline{P}^{[\mathfrak{L}]}\left(\mathscr{W};\eta_\mathrm{R}\right)$ only. It can be observed that with an increasing field order, the upper bound on $\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\}$ or $\mu\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\}$, respectively, converges quickly to the true value, so that it is sufficiently accurate to use only $\overline{P}^{[\mathfrak{L}]}\left(\mathscr{W};\eta_\mathrm{R}\right)$ for non-binary sparse random ensembles.

In Figure 3.7 the expected absolute reception overhead of the standard random ensemble set and that of ensemble set A is depicted as well as the respective upper bounds as dashed lines. For the standard random ensemble, $\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\}$ is only slightly smaller than for the sparse random ensemble set A. Decreasing the average row weight further, however, leads to a notable rise of $\mathrm{E}^{[\mathfrak{L}]}\{\eta_\mathrm{R}\}$.

3.5.3 Concentrated Ensembles

Like the sparse random ensemble, the ensemble with a concentrated row weight distribution has a near-optimal erasure correction performance. It is interesting to see that using an equivalent parametrisation of the two ensembles, i.e. equal average row weight and input size, their erasure correction performance is virtually the same. So, accordingly parametrised concentrated ensembles yield essentially the same erasure probabilities as the ones depicted in Figures 3.3 to 3.7.

Since a general expression for the exact word erasure probability as a function of the row weight distribution is not known, the upper bound on the residual word



(a)

(b)

Figure 3.8: Word erasure probabilities of purely concentrated ensembles (measured) and expurgated random ensembles (3.14), respectively, for absolute symbol reception overheads from 0 to 10 over $\mathbb{F}_2$ and $\mathbb{F}_4$ for very small input sizes. The row weights of the purely concentrated ensembles are $d = k/q$. Note that for the binary ensembles only odd values are used for d.

erasure probability (3.18) constitutes a viable objective function to find the optimal row weight distribution. Without a constraint on the average row weight, numerical optimisation yields the row weight distribution of the expurgated random ensemble. However, the difference in the objective function using the optimal distribution to the case with an equivalently parametrised concentrated distribution is essentially zero. The existence of these two special row weight distributions, which are by their appearance so very different, suggests that there exist many row weight distributions that are close to optimal. While this assumption can be easily verified by generating row weight distributions that have some (here not further specified) likeness to the two above mentioned distributions, the more daring supposition that any row weight distribution might deliver close to optimal results does not hold.

Despite the existence of many close to optimal row weight distributions, only the two types random and concentrated shall be discussed in the following, as there is enough reason to assume that the characteristics of all other close to optimal ensembles are mixtures of these two cases. In order to illustrate this virtual equality of the erasure correction performance of the two ensemble types, the word erasure

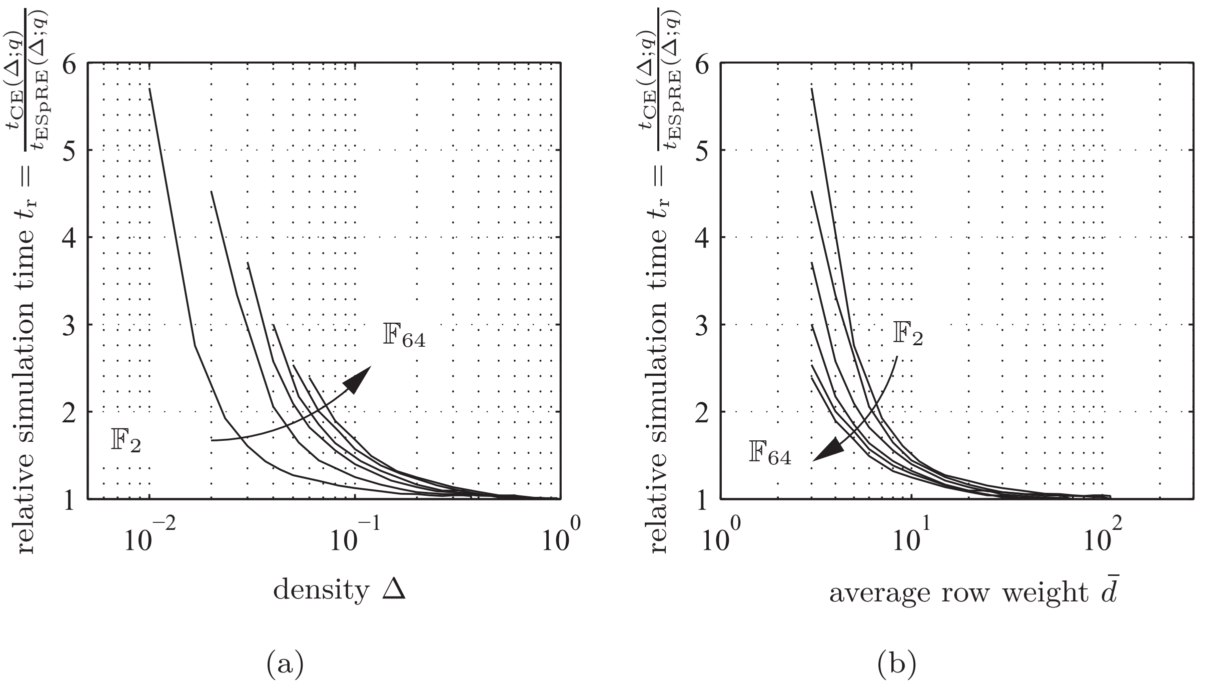


Figure 3.9: Simulation times of concentrated ensembles relative to those of expurgated sparse random ensembles of the same average row weight or density at an inverse reception code rate $\gamma_R \approx 1.04$. In (a) the simulation times are depicted as a function of the density, while in (b) they are given in terms of the average row weight $\bar{d}$. The ensembles have input sizes $k = \frac{300}{\mathrm{ld}(q)} = \frac{300}{\mu} \in \{300, 150, \ldots, 50\}$.

probability is depicted in Figure 3.8 for purely concentrated ensembles and for expurgated random ensembles for very small input sizes, where a difference is still visible. Considering the binary ensembles, hardly any difference can be observed already for input sizes as small as $k = 14$ and it vanishes quickly as k grows, with the expurgated random ensemble being always infinitesimally better than the concentrated one. For non-binary ensembles this strong congruence begins at even smaller input sizes.

Moreover, this congruence can also be observed for sparse random ensembles of the same average row weights as the concentrated ones. Adding a constraint on the average row weight such that it is lower than approximately $k(1 - 1/q)$, i.e. lower than the average row weight of the expurgated random ensemble, the optimal row weight distribution tends more and more towards the concentrated distribution as the average row weight approaches one. In fact, for the extreme case of an average row weight of one the expurgated sparse ensemble and the concentrated ensemble are identical. Nevertheless, the difference in the objective function for the two cases is again so marginal that it is hardly worth the effort.

Judging from the erasure correction performance both types of ensembles, i.e. expurgated random or concentrated, are equally good, and with densities close to

$1 - 1/q$, the decoding complexities are also almost equal. However, the decoding complexities differ significantly for lower densities or average row weights as can be observed in Figure 3.9. The decoding complexities have been estimated by measuring the simulation times for ensembles of input sizes $k = \frac{300}{\mathrm{ld}(q)} = \frac{300}{\mu} \in \{300, 150, \ldots, 50\}$ over the respective fields.

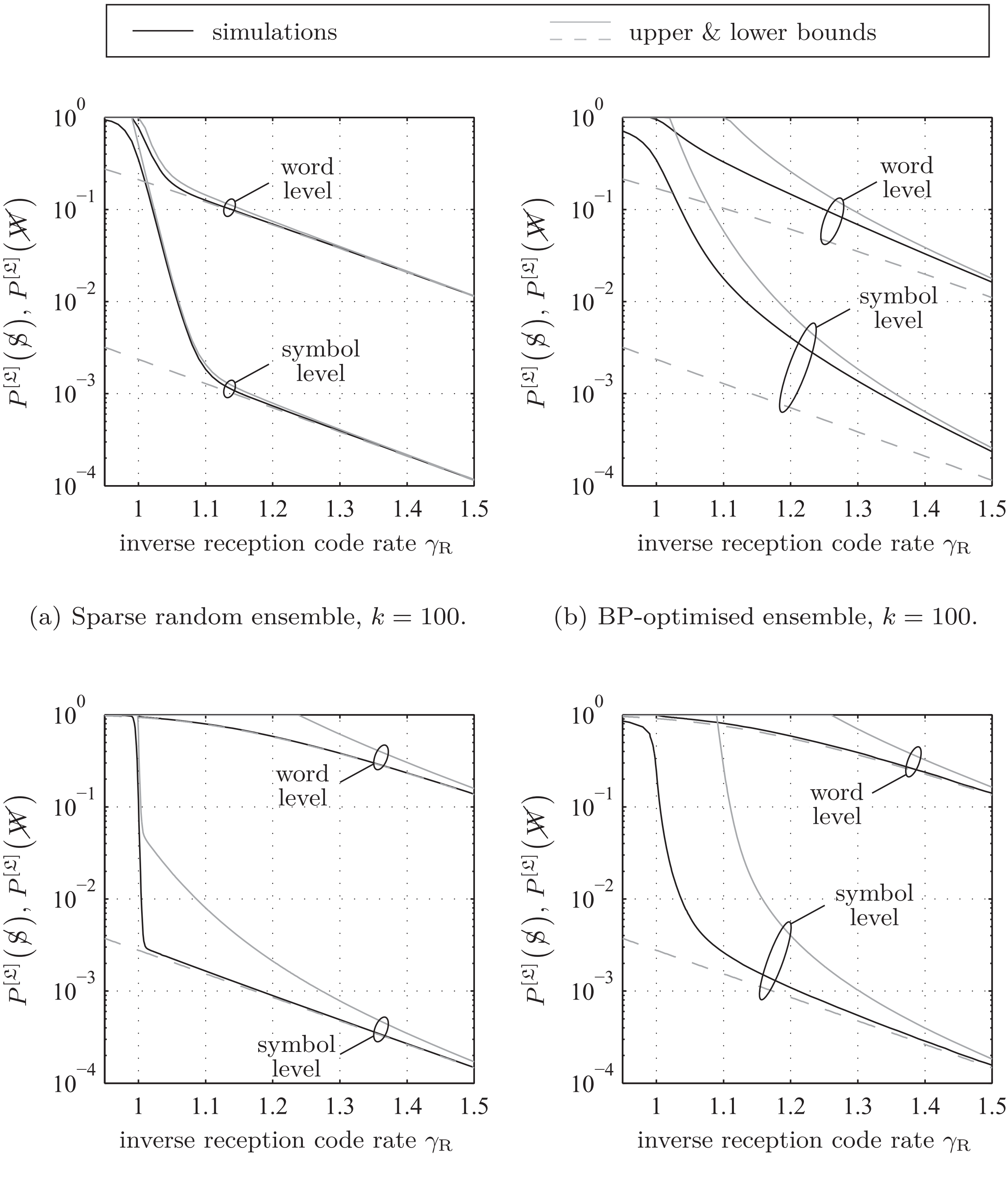
In Figure 3.9 the simulation times $t_{\mathrm{CE}}(\Delta; q)$ of the purely concentrated ensembles are depicted relatively to the times $t_{\mathrm{ESpRE}}(\Delta; q)$ of the equivalently parametrised expurgated sparse random ensembles. At lower average row weights, the random ensembles profit from the increasing probability of occurrence of very low weight rows which speed up the pivoting process. Most helpful are rows of weight one, as these can be directly used for back substitution. The average row weight, however, should not be decreased too far, since for $\bar{d} \lesssim \ln k$ the ensembles begin to degenerate. So, besides their trivial parametrisation, concentrated ensembles do not show a practical advantage over (sparse) random ensembles. Therefore, concentrated ensembles will not be further considered in the following.

3.5.4 A BP-Optimised Ensemble under ML Decoding

A row weight distribution often used in the literature is given by (2.8) from [Sho06]. As it has been optimised for BP decoding (cf. Sections 2.1.5 and Section 2.1.6) and for a larger input size it shall just serve as a contrasting example to the distributions used so far. The erasure correction performance of two binary ensembles with this distribution is depicted in Figure 3.10(b) and Figure 3.10(d) for the two input sizes $k = 100$ and $k = 1000$ in terms of simulated erasure rates together with the respective upper and lower bounds on the erasure probability again both on word and on symbol level. For comparison, the performance of sparse random ensembles with the same input sizes and average row weight $\bar{d} = 5.87$ is shown in Figure 3.10(a) and Figure 3.10(c). This parametrisation ensures that the ensemble pairs with equal input size have equal symbol erasure floors and virtually equal word erasure floors.

The two regions waterfall and erasure floor of the short BP-optimised ensemble are not very distinctive, neither in the upper bound nor in the simulated erasure rates, whereas for the long BP-optimised ensemble the two regions become more pronounced, particularly for the simulated erasure rates. In general, these ensemble have a worse erasure correction performance than the corresponding sparse random ensembles. However, the long BP-optimised ensemble has already a competitive performance and it becomes even better with larger k.

Unfortunately, the bounds are less tight for the BP-optimised ensembles and the simulated erasure rates converge slower to their bounds than it is the case for the sparse random ensembles. For this reason, the erasure correction performance of such ensembles cannot be predicted very accurately by the upper bounds as it is possible for that of sparse random ensembles.



(a) Sparse random ensemble, $k = 100$.

(b) BP-optimised ensemble, $k = 100$.

(c) Sparse random ensemble, $k = 1000$.

(d) BP-optimised ensemble, $k = 1000$.

Figure 3.10: Upper and lower bounds on the residual erasure probabilities on word and on symbol level as well as simulated erasure rates for two types of binary LT code ensembles and for two different input sizes $k = 100$ and $k = 1000$.
Code details:
On the left: sparse random ensembles with $\bar{d} = 5.87$.
On the right: BP-optimised ensembles with $\bar{d} = 5.87$.

Despite the inferior erasure correction performance, it should be kept in mind that optimising for BP decoding yields ensembles that have a very low decoding complexity. This particularly pays off for large k, where the decoding of sparse random ensembles becomes much more complex than the decoding of BP-optimised ensembles. Moreover, the erasure correction performance of BP-optimised ensembles becomes comparable to that of equivalently parametrised sparse random ensembles, which can be observed in Figures 3.10(d) and 3.10(c).

The BP-optimised ensemble in Figure 3.10(b) for instance, can be ML decoded in approximately 42.2 % of the time that is required to ML decode the sparse random ensemble in Figure 3.10(a). But the significantly better performance of the sparse random ensemble definitely justifies the additional complexity. On the other hand, the BP-optimised ensemble in Figure 3.10(d) is ML-decodable[3] in approximately 4.75 % of the time required for the sparse random ensemble in Figure 3.10(c). In the latter case, the slightly superior erasure correction performance of the sparse random ensemble does not necessarily justify the complexity overhead. Thus, for large k, BP-optimised ensembles should be used, but for small to medium k, sparse random ensembles are a much better choice.

Another detail of the performance curves shall also be brought into focus. As the average row weight $\bar{d}$ has been kept constant, the density of the longer codes, i.e. with $k = 1000$, is much lower than that of the short codes. In particular $\bar{d}$ is way lower than $\ln k$, which is the approximate size where the codes start to degenerate. Here, the term degenerated is used rather informally, whenever the word erasure probability does not exhibit a noticeable waterfall. Moreover, the word erasure probabilities of the longer codes, unlike the short codes, are extremely close to their lower bounds, since due to the low density, decoding failures arise mostly from unconnected input nodes, i.e. from all-zero columns in the decoding matrix. The larger the distance of the actual erasure probability from the lower bound on word level, as in the case of the short BP-optimised code, the more linearly dependent not all-zero columns exist in the decoding matrix.

The Probability of Exactly j Unconnected Input Nodes

The lower bound on word level is nothing but the sum of the probabilities $P_\mathrm{e}(j)$ of having exactly j input nodes that are not connected to any output node, for which a closed form expression has been derived in Section 3.3.5. These probabilities together with the lower bound on word level are depicted in Figure 3.11 for the previous four code ensembles. It can be observed that $P_\mathrm{e}(j)$, with $j \geq 2$, for the sparse random ensembles is smaller than for the BP-optimised ensembles. With these probabilities it is possible to gain some more insight into the respective codes,

[3] Note that BP decoding of the BP-optimised ensemble yields almost the same result as ML decoding, both in terms of residual erasure probability as well as complexity.

as $P_\mathrm{e}(j)$ serves as a more general lower bound, i.e. the probability that there are j or more residual symbol erasures in an information word is greater than or equal to $P_\mathrm{e}(j)$. Although this general lower bound looses tightness with increasing j (not shown here), it can at least be used as an indicator for the precodability of an LT code ensemble.

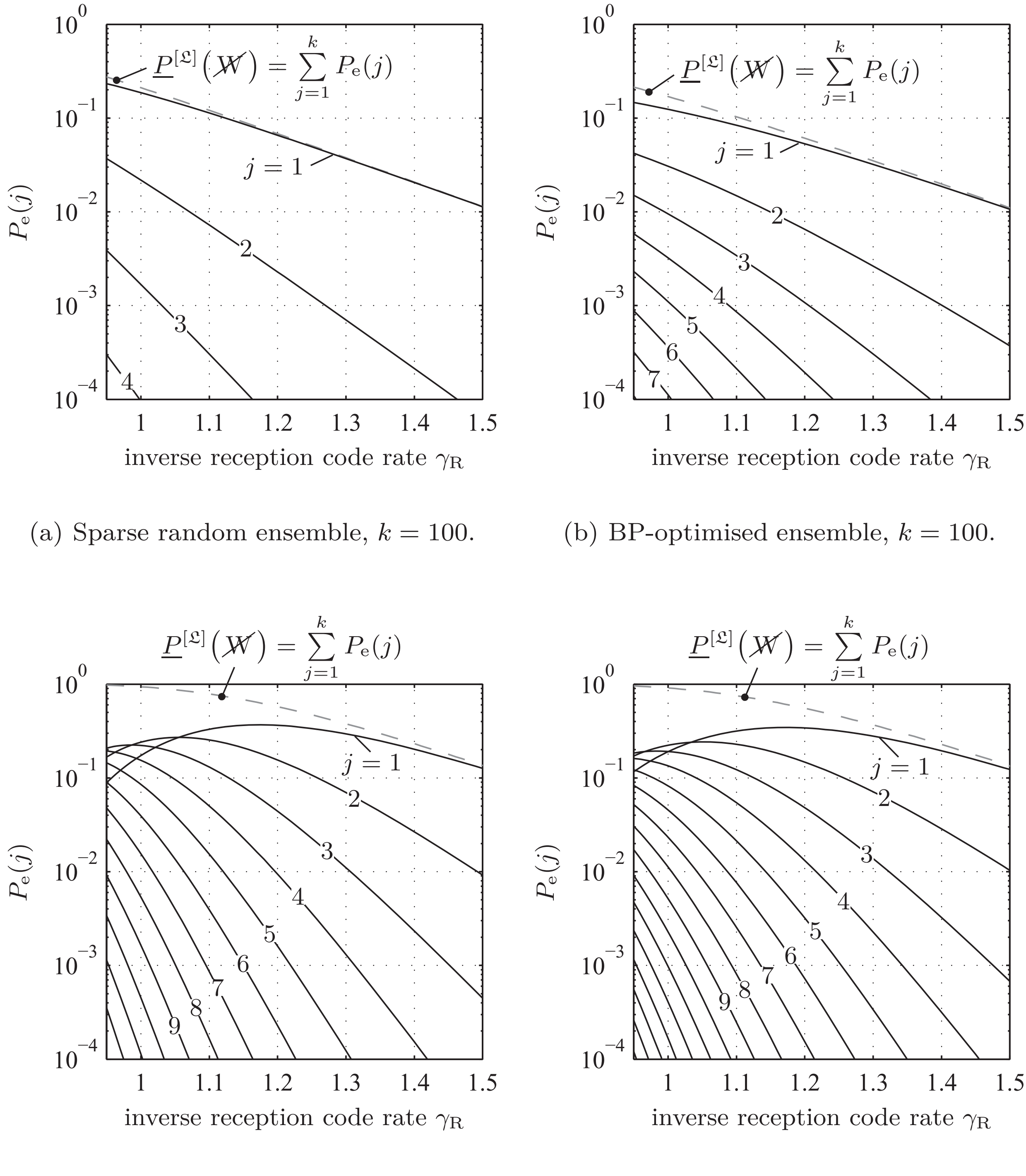
3.6 Computational Complexity

In Section 3.5.1 the erasure correction performance of standard random ensembles has been assessed for different field orders. To this end, a fairness constraint has been introduced, such that the number of input bits is kept constant when changing the field order. Although the unconstrained approach from the literature yields enormous erasure performance gains for higher field orders, as depicted in Figure 3.2(a), these come at the price of a significant increase in computational complexity [LPC10]. While in the unconstrained approach the dimensions of the underlying systems of linear equations are equal over different fields (the number of symbols is kept constant), the operations over higher field orders are computationally more expensive than those over lower ones. A yet more significant complexity penalty is due to the density of the standard random ensembles which grows quickly from 0.5 in the binary case and approaches one as the field order increases.

3.6.1 The Constrained Standard Random Ensemble

By introducing the constraint on the number of bits per information word, apart from enabling a fair comparison, the erasure correction performance still increases with the field order, though not as fast as without this constraint, as can be seen in Figure 3.2(b). The choice of a higher field order, however, does not only lead to a slight *improvement* of the erasure correction performance, but more importantly it comes with a significantly *lower* computational complexity. ML decoding over $\mathbb{F}_2$ has a complexity of $\mathcal{O}(k_\mathrm{B}^3)$ per information word. Using higher order Galois fields while keeping the number k_B of input bits constant, the input size $k = k_\mathrm{B}/\mathrm{ld}(q) = k_\mathrm{B}/\mu$ in terms of symbols over $\mathbb{F}_q$ decreases with the number of bits per symbol, so that in total fewer though a little more complex computation steps are necessary, i.e. $\mathcal{O}(\beta_\mu k^3)$ with $\beta_\mu > 1$.

There exist several optimised methods for operations over higher order Galois fields. In [GMS08], for instance, an optimised approach for multiplications over $\mathbb{F}_q$ has been proposed. Since computer architectures are generally based on byte operations, ensembles over $\mathbb{F}_{256}$ shall be considered exemplarily, i.e. $\mu = 8$. When using, e.g. the 'Log/Antilog Optimized' technique for the multiplications over $\mathbb{F}_{256}$ and assuming three table lookups and one add operation per multiplication (cf. [GMS08, Table 1]), the complexity factor β_8 is coarsely estimated as $\beta_8 = 4$. With this assumption, the complexity of the ensemble over $\mathbb{F}_{256}$ is approximately



(a) Sparse random ensemble, $k = 100$.

(b) BP-optimised ensemble, $k = 100$.

(c) Sparse random ensemble, $k = 1000$.

(d) BP-optimised ensemble, $k = 1000$.

Figure 3.11: Probability $P_\mathrm{e}(j)$ (cf. (3.53)) of exactly j input nodes being not connected to any of the $k\gamma_\mathrm{R}$ output nodes as well as the lower bound on word level, i.e. $\underline{P}^{[\mathfrak{L}]}\big(\cancel{W}\big) = \sum_{j=1}^{k} P_\mathrm{e}(j)$ (cf. (3.48)), for two types of binary LT code ensembles and for two different input sizes $k = 100$ and $k = 1000$.
Code details:
On the left: sparse random ensembles with $\bar{d} = 5.87$.
On the right: BP-optimised ensembles with $\bar{d} = 5.87$.

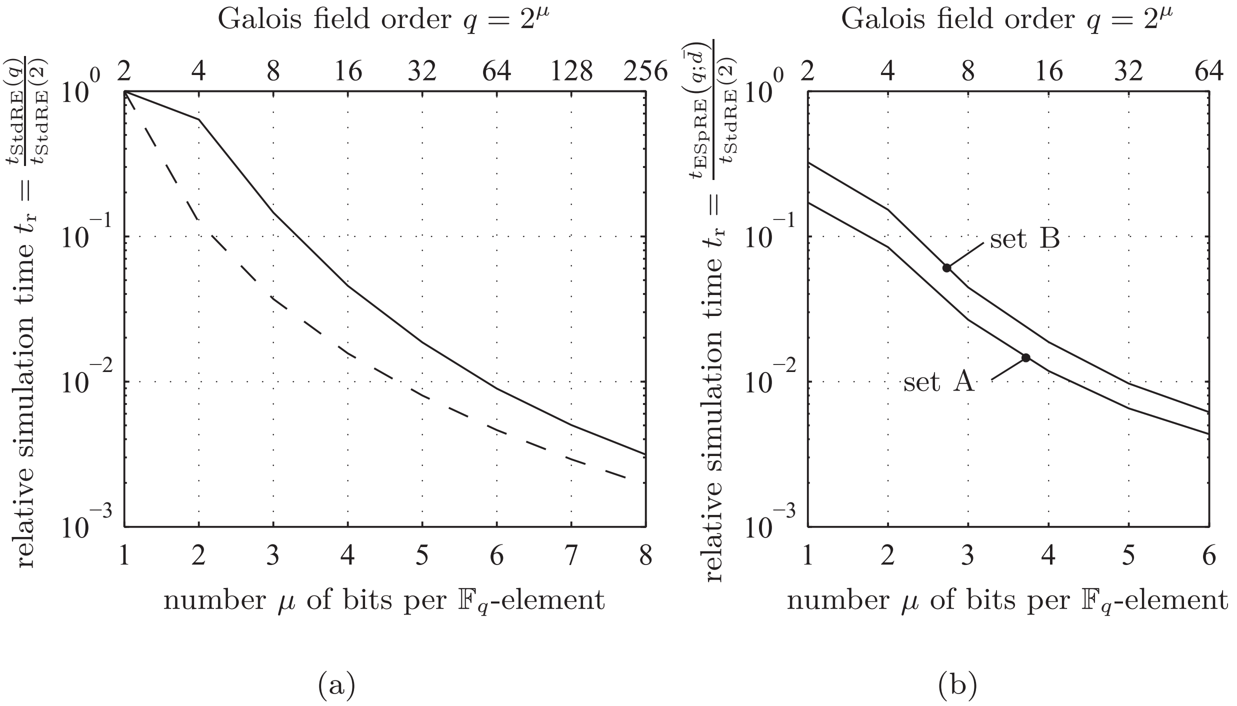


Figure 3.12: Relative simulation times of standard and sparse random ensembles as a function of the Galois field order.

(a) Relative simulation times of the standard random ensembles from Figure 3.2(b) with input sizes $k = \frac{840}{\mathrm{ld}(q)} = \frac{840}{\mu} \in \{840, 420, \dots, 105\}$ and $\mu \in \{1, 2, \dots, 8\}$ at an inverse reception code rate $\gamma_{\mathrm{R}} \approx 1.01$. The simulation times $t_{\mathrm{StdRE}}(q)$ are given relative to the simulation time $t_{\mathrm{StdRE}}(2)$ of the binary code. The dashed line ($t_{\mathrm{r}} = \mu^{-3}$) indicates the order of complexity (the Gaussian elimination algorithm has a complexity of $\mathcal{O}(k^3)$ per information word).

(b) Relative simulation times of the sparse random ensemble sets A ($\bar{d} = 10$) and B ($\bar{d} = 15$) from Figure 3.5 with input sizes $k = \frac{300}{\mathrm{ld}(q)} = \frac{300}{\mu} \in \{300, 150, \dots, 50\}$ and $\mu \in \{1, 2, \dots, 6\}$ at an inverse reception code rate $\gamma_{\mathrm{R}} \approx 1.04$. The simulation times are given relative to the simulation time $t_{\mathrm{StdRE}}(2)$ of the binary standard random ensemble, whose density is $\Delta = 0.5$.

$\mathcal{O}((\beta_8 k)^3) = \mathcal{O}((4 \cdot \frac{k_{\mathrm{B}}}{8})^3) = \mathcal{O}(\frac{1}{8} k_{\mathrm{B}}^3)$. Consequently, with optimised Galois field arithmetic implementations and eventually also with smart ML decoding algorithms, the computational complexity per information word can be decreased even further as the field order q increases.

In Figure 3.12(a) relative simulation times of the standard random ensembles from Figure 3.2(b) are depicted. The simulation times have been measured at an inverse reception code rate $\gamma_{\mathrm{R}} \approx 1.01$ and they are given relative to that of the binary ensemble. The dashed line ($t_{\mathrm{r}} = \mu^{-3}$) indicates the order of complexity when assuming that binary and non-binary operations are equally complex. In that case, the Gaussian elimination algorithm would have a complexity of $\mathcal{O}(k^3) = \mathcal{O}((\frac{k_\mathrm{B}}{\mu})^3)$ per information word. For these simulations, the complexity factors β_μ are greater than one but less than four, i.e. the achieved simulation time speedups are significantly better than estimated in the previous paragraph. Of course, the actual factors β_μ strongly depend on the implementation.

3.6.2 The Sparse Random Ensemble

However, the complexity of the standard random ensembles is often still too high for practical purposes and the property that these ensembles exhibit no erasure floor is usually not required. Instead, the erasure correction performance of their sparse counterparts is almost as good in the waterfall region but they can be processed with far lower computational costs. The relative simulation time of the two example ensemble sets from Figure 3.5 over $\mathbb{F}_2$ to $\mathbb{F}_{64}$ with $\bar{d} = 10$ (ensemble set A) and $\bar{d} = 15$ (ensemble set B) and with 300 input bits is shown in Figure 3.12(b). It is again given relative to that of the binary standard random ensemble, i.e. with $\Delta = 0.5$. Also in the sparse random case it can be observed that the computational complexity decreases with an increasing q, while the erasure correction performance improves at the same time.

In Figure 3.13(a) the relative simulation times of sparse random ensembles with 300 input bits over $\mathbb{F}_2$ to $\mathbb{F}_{64}$ are depicted for the complete range of densities measured at an inverse reception code rate of $\gamma_\mathrm{R} \approx 1.04$. The diamond markers indicate the standard random ensembles which have a density of $\Delta = 1 - 1/q$. Apart from the favourable speedup effect of higher order Galois fields that can be observed, it is worthwhile to note that the simulation time curves have a very characteristic shape. For very low densities a small variation of the density has a dramatic influence on the simulation times, while at the high density end the curve flattens and a variation of the density has only a very limited effect on the simulation times. So in order to save complexity it makes sense to choose the highest supported field order together with the lowest density that still fulfils the erasure correction and granularity requirements.

Although Figure 3.13(a) gives a good overview of the complexity as a function of the density, ensembles over different fields but with the same density should not be directly compared, since they have different erasure correction properties. To even out this discrepancy it is more advantageous to plot the relative simulation times of ensemble sets, e.g. as in Figure 3.13(b) for ensemble sets with equal word erasure floors. The density of ensembles over one field size is taken as the reference, here the one over $\mathbb{F}_{64}$. Then, the densities $\Delta^\star$ of the dependent ensembles in the respective set can be computed via (3.64) or (3.67).

With Figure 3.13(b) it is now fairly easy to compare the complexity of ensemble sets with equal word erasure floor as they have the same abscissa. The simulation

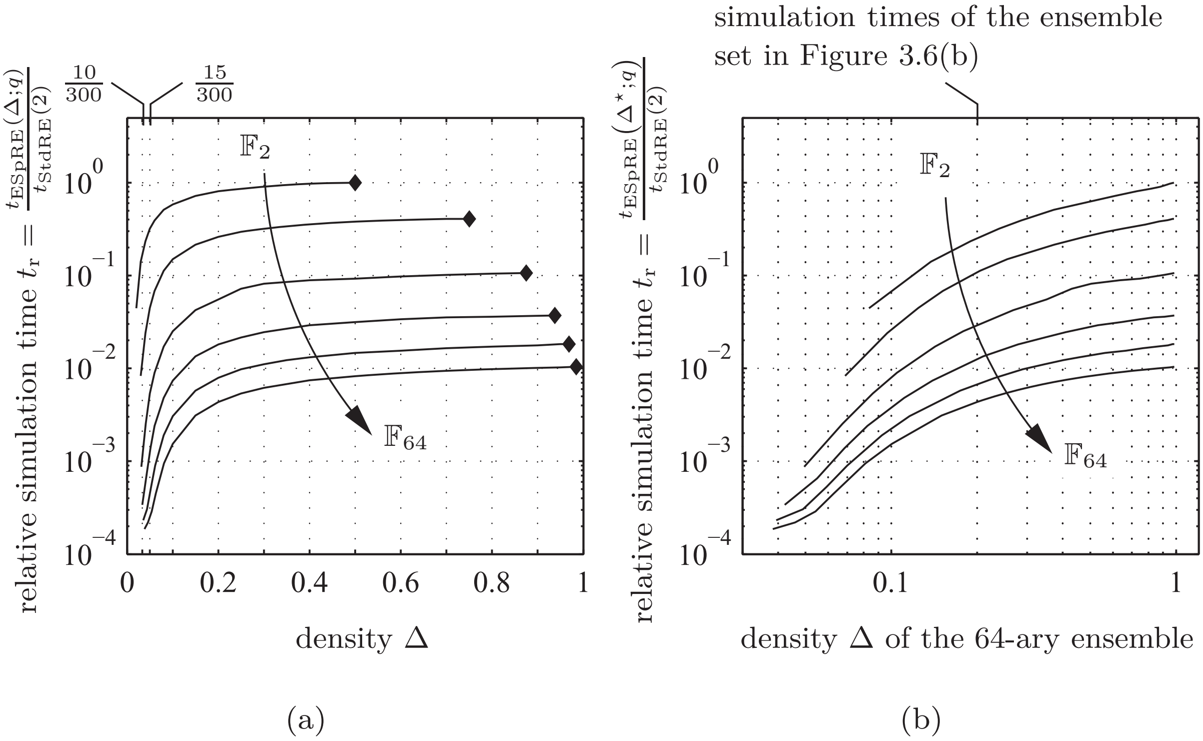


Figure 3.13: Relative simulation times of standard and sparse random ensembles with input sizes $k = \frac{300}{\mathrm{ld}(q)} = \frac{300}{\mu} \in \{300, 150, \ldots, 50\}$ and $\mu \in \{1, 2, \ldots, 6\}$ at an inverse reception code rate $\gamma_{\mathrm{R}} \approx 1.04$. The times are given relative to the simulation time $t_{\mathrm{StdRE}}(2)$ of the binary standard random ensemble, i.e. with density $\Delta = 0.5$.

(a) The simulation times $t_{\mathrm{ESpRE}}(\Delta; q)$ are given as a function of Δ. The diamond markers indicate the standard random ensembles over the respective field. These have the maximum considered density of $\Delta = 1 - 1/q$ for the respective field order q.

(b) The simulation times $t_{\mathrm{ESpRE}}(\Delta^\star; q)$ are given as a function of the density of the 64-ary ensembles, which is taken as the reference density to construct ensemble sets with the same word erasure floors as the 64-ary ensembles. The densities $\Delta^\star$ of the other ensembles are obtained from Δ by means of (3.67). In this plot, the points on the 6 curves with the same abscissa correspond to ensembles sets with equal word erasure floors and thus the complexity of such ensemble sets can be easily compared.

times of the exemplary ensemble set from Figure 3.6(b) can be found at a density $\Delta = 0.2$ of the 64-ary ensemble as indicated in the plot. Merely by changing the Galois field order from binary to 64-ary the complexity drops by almost two decades, while the ensembles in this set still have the same erasure correction performance in the word erasure floor region and the non-binary ones have even a superior performance in the waterfall.

3.7 Conclusions

General and some special LT code ensembles have been analysed in the present chapter with respect to their erasure resilience under optimal decoding. The derived exact erasure probabilities or the bounds thereon constitute useful tools for the analysis and the design of these ensembles. In this chapter, the derivations have been presented in great detail, while in the following chapters bounds for the same purpose but under different conditions are merely stated without further proof, as the general procedures are similar to the ones described here.

A particular emphasis has been laid on the random ensembles. The near-optimal sparse random ensembles have turned out as an excellent alternative to the optimal standard random ensemble due to the almost indistinguishable erasure correction performance in the practically relevant region of low reception overheads, i.e. in the waterfall region, at a significantly lower computational cost. The generalisation to and the usage of Galois fields of higher order has also proven beneficial resulting in a lower residual erasure probability in the waterfall region and at the same time in a considerably decreased computational complexity. By the introduction of well-defined code ensemble sets, i.e. ensembles over fields of different order with a certain property in common, such as an equal erasure floor on word level, a fair comparison of ensembles over different fields is facilitated.

Chapter 4

Conventionally Systematic LT Code Ensembles

The erasure correction performance of structured ensembles depends on the erasure probability ϵ of the channel as already described in Section 2.2. Particularly for conventionally systematic ensembles it is worthwhile to examine their performance as a function of the channel [Yua12], as it strongly depends on the used row weight distribution whether the conventionally systematic prefix is beneficial or not.

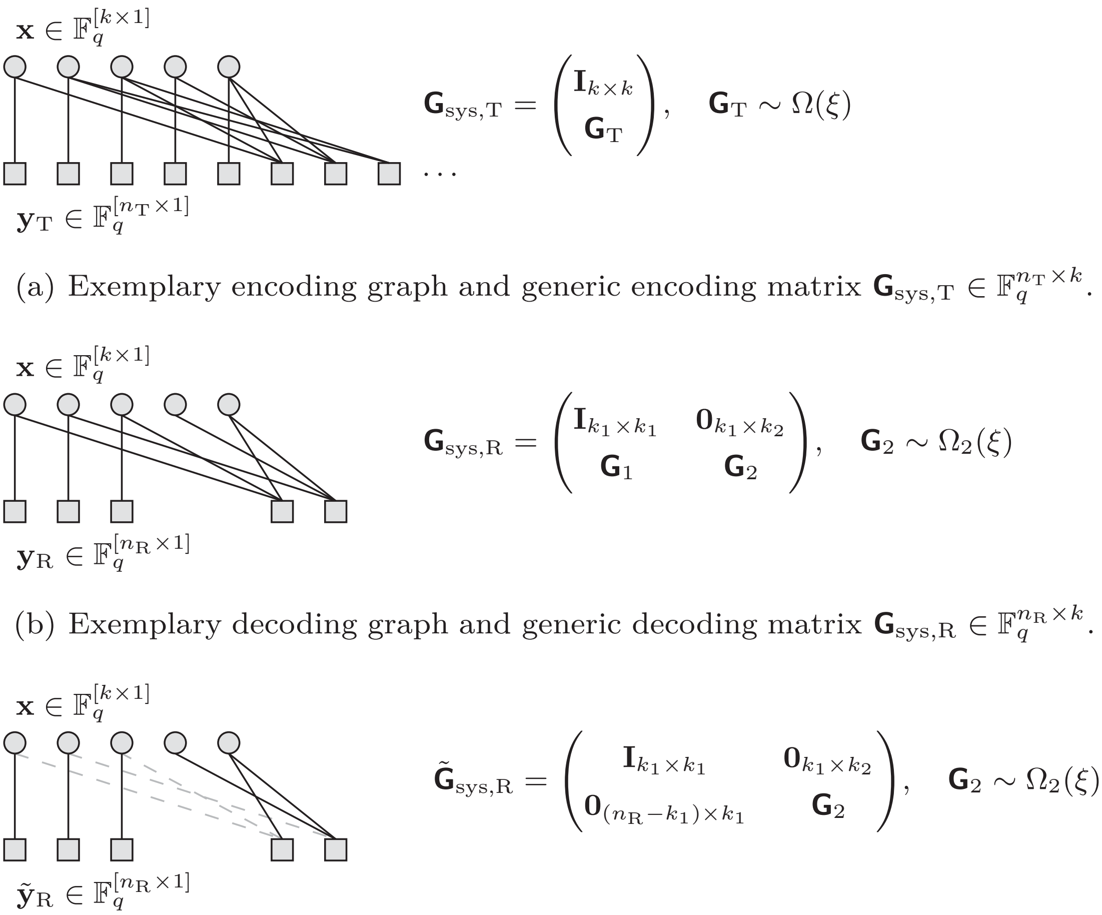
The prefix of a conventionally systematic LT code ensemble consists of the $k \times k$ identity matrix $\mathbf{I}_{k\times k}$, i.e. the first k encoded symbols are equal to the k data symbols. Of the k transmitted systematic encoded symbols only k_1 are received, i.e. $k_2 = k - k_1$ systematic encoded symbols are erased. In total n_R encoded symbols are received of which the non-systematic encoded symbols are randomly generated according to a given row weight distribution $\Omega(\xi)$. At the receiver, the matrix

$$\mathbf{G}_{\mathrm{sys,R}} = \begin{pmatrix} \mathbf{I}_{k_1\times k_1} & \mathbf{0}_{k_1\times k_2} \\ \mathbf{G}_1 & \mathbf{G}_2 \end{pmatrix} \in \mathbb{F}_q^{n_\mathrm{R}\times k} \tag{4.1}$$

is used to decode the remaining input symbols. Without loss of generality, the input symbols and thus the columns of matrix $\mathbf{G}_{\mathrm{sys,R}}$ can be permuted to place the received systematic symbols on the first k_1 positions.

As depicted in Figure 4.1, this results in an identity matrix of size $k_1 \times k_1$ in the upper left part of $\mathbf{G}_{\mathrm{sys,R}}$ and an all-zero matrix of size $k_1 \times k_2$ in the upper right part. The following $n_\mathrm{R} - k_1$ rows are generated according to the distribution $\Omega(\xi)$. These rows can be split into two submatrices $\mathbf{G}_1$ and $\mathbf{G}_2$ of size $(n_\mathrm{R} - k_1) \times k_1$ and $(n_\mathrm{R} - k_1) \times k_2$, respectively. Using only backward insertion of the known systematic symbols, the matrix $\mathbf{G}_{\mathrm{sys,R}}$ can be transformed to

$$\tilde{\mathbf{G}}_{\mathrm{sys,R}} = \begin{pmatrix} \mathbf{I}_{k_1\times k_1} & \mathbf{0}_{k_1\times k_2} \\ \mathbf{0}_{(n_\mathrm{R}-k_1)\times k_1} & \mathbf{G}_2 \end{pmatrix} \tag{4.2}$$


$\mathbf{x} \in \mathbb{F}_q^{[k\times 1]}$
...
$\mathbf{y}_\mathrm{T} \in \mathbb{F}_q^{[n_\mathrm{T}\times 1]}$
$\mathbf{x} \in \mathbb{F}_q^{[k\times 1]}$
$\mathbf{y}_\mathrm{R} \in \mathbb{F}_q^{[n_\mathrm{R}\times 1]}$
$\mathbf{x} \in \mathbb{F}_q^{[k\times 1]}$
$\tilde{\mathbf{y}}_\mathrm{R} \in \mathbb{F}_q^{[n_\mathrm{R}\times 1]}$

$$\mathbf{G}_{\mathrm{sys,T}} = \begin{pmatrix} \mathbf{I}_{k\times k} \\ \mathbf{G}_\mathrm{T} \end{pmatrix}, \quad \mathbf{G}_\mathrm{T} \sim \Omega(\xi)$$

(a) Exemplary encoding graph and generic encoding matrix $\mathbf{G}_{\mathrm{sys,T}} \in \mathbb{F}_q^{n_\mathrm{T}\times k}$.

$$\mathbf{G}_{\mathrm{sys,R}} = \begin{pmatrix} \mathbf{I}_{k_1\times k_1} & \mathbf{0}_{k_1\times k_2} \\ \mathbf{G}_1 & \mathbf{G}_2 \end{pmatrix}, \quad \mathbf{G}_2 \sim \Omega_2(\xi)$$

(b) Exemplary decoding graph and generic decoding matrix $\mathbf{G}_{\mathrm{sys,R}} \in \mathbb{F}_q^{n_\mathrm{R}\times k}$.

$$\tilde{\mathbf{G}}_{\mathrm{sys,R}} = \begin{pmatrix} \mathbf{I}_{k_1\times k_1} & \mathbf{0}_{k_1\times k_2} \\ \mathbf{0}_{(n_\mathrm{R}-k_1)\times k_1} & \mathbf{G}_2 \end{pmatrix}, \quad \mathbf{G}_2 \sim \Omega_2(\xi)$$

(c) Exemplary transformed decoding graph and generic transformed decoding matrix $\tilde{\mathbf{G}}_{\mathrm{sys,R}} \in \mathbb{F}_q^{n_\mathrm{R}\times k}$. The dashed grey lines represent removed edges.

Figure 4.1: A conventionally systematic LT code.

in which the submatrix $\mathbf{G}_1$ has been turned into an all-zero matrix while submatrix $\mathbf{G}_2$ is left unchanged. Since the first k_1 input symbols are already known, the erasure properties of the LT code are fully determined by submatrix $\mathbf{G}_2$. Therefore, the performance of a conventionally systematic LT code of length k under ML decoding is first analysed under the condition that $n_\mathrm{R} = k\gamma_\mathrm{R}$ encoded symbols are received and that k_2 of the k systematic encoded symbols are erased on the SEC, so initially expressions for the bounds on the conditional symbol and word erasure probabilities $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{S}}\,\big|\,k_2\big)$ and $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\,\big|\,k_2\big)$ under ML decoding have to be found. Note that for actually determining the remaining input symbols, the received symbols $\mathbf{y}_\mathrm{R}$ need to be updated according to the matrix transformations. For the determination of the erasure correction properties, however, it is sufficient to consider only the decoding matrix.

Although the transformed $n_\mathrm{R} - k_1$ rows in (4.2) obviously do not follow the row weight distribution $\Omega(\xi)$ anymore, the distribution $\Omega_2(\xi)$ for submatrix $\mathbf{G}_2$, which is required for assessing the erasure correction properties of the given LT code ensemble, can be obtained from the original distribution as will be shown subsequently. Sometimes it is convenient to express the number of rows of the considered matrices in terms of the inverse reception code rate γ_{R}, the input size k as well as k_2, the number of erased systematic encoded symbols. Thus, $\mathbf{G}_2$ has $n_{\mathrm{R}} - k_1 = k(\gamma_{\mathrm{R}} - 1) + k_2$ rows.

4.1 The Row Weight Distribution of a Submatrix

Lemma 4.1. Given n_{R} received output nodes and k_2 erasures in the systematic positions, the row weight distribution of submatrix $\mathbf{G}_2 \in \mathbb{F}_q^{(k(\gamma_{\mathrm{R}}-1)+k_2)\times k_2}$ is

$$\begin{aligned}\Omega_2(\xi) &= \sum_{d_2 \in \mathcal{D}_2} \Omega_{2,d_2} \xi^{d_2} &(4.3)\\ &= \sum_{d_2 \in \mathcal{D}_2} \left(\sum_{d \in \mathcal{D}} \Omega_d \cdot \frac{\binom{k-k_2}{d-d_2}\binom{k_2}{d_2}}{\binom{k}{d}} \right) \xi^{d_2}, &(4.4)\end{aligned}$$

where the non-systematic output node degree d_2 denotes the number of edges of an output node that are connected to input nodes without a systematic description at the receiver.

Proof. The coefficients of the distribution $\Omega_2(\xi)$ are given by

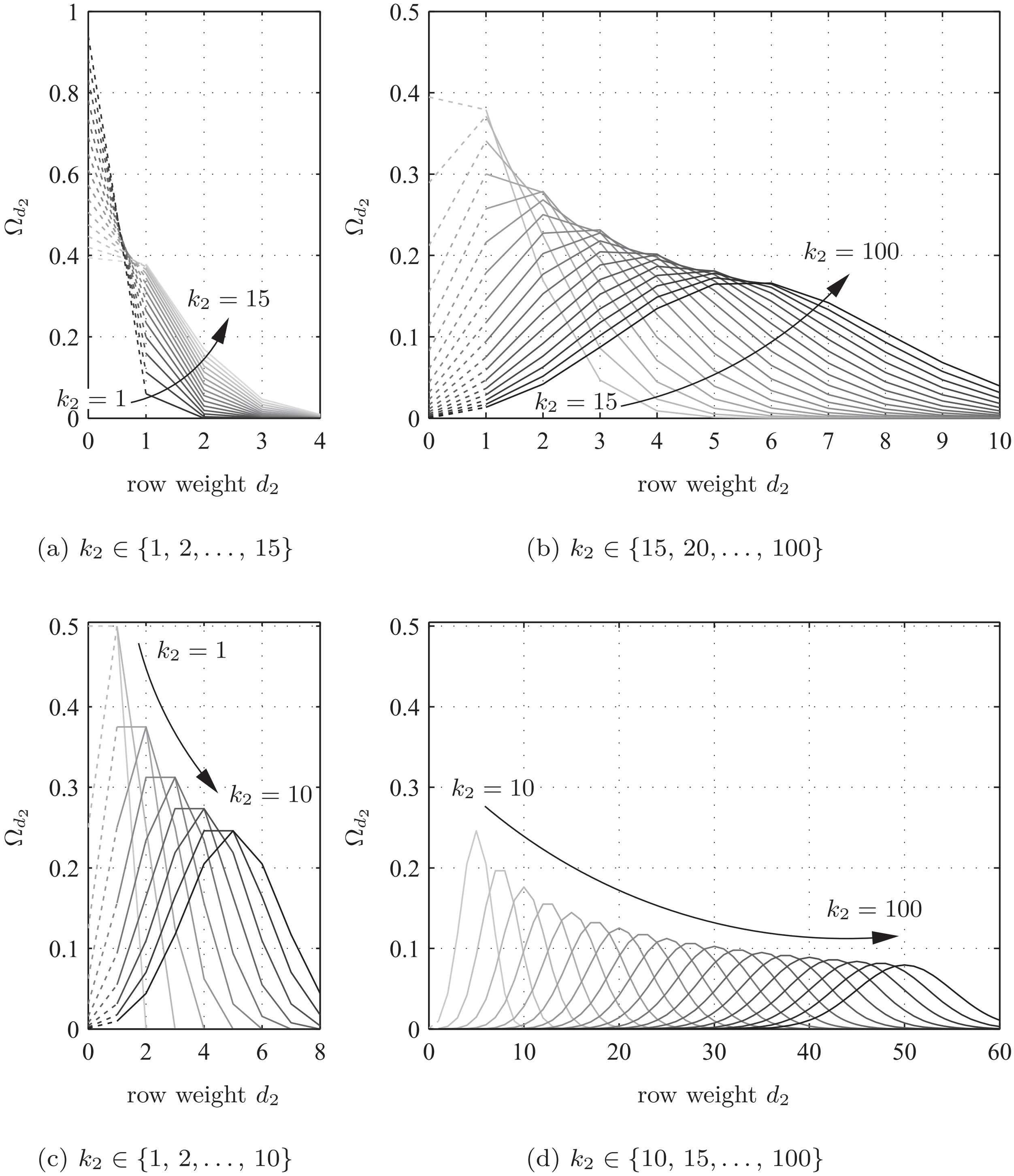
$$\begin{aligned}\Omega_{2,d_2} &= \Pr\{\mathsf{d}_2 = d_2\}\\ &= \sum_{d \in \mathcal{D}} \Pr\{\mathsf{d} = d\} \cdot \Pr\{\mathsf{d}_2 = d_2 \mid \mathsf{d} = d\}\\ &= \sum_{d \in \mathcal{D}} \Omega_d \cdot \Pr\{\mathsf{d}_2 = d_2 \mid \mathsf{d} = d\} &(4.5)\end{aligned}$$

with the hypergeometric distribution

$$\Pr\{\mathsf{d}_2 = d_2 \mid \mathsf{d} = d\} = \frac{\binom{k-k_2}{d-d_2}\binom{k_2}{d_2}}{\binom{k}{d}}. \qquad (4.6)$$

Equation (4.6) is due to the fact that, given an output node of degree d, there are $\binom{k_2}{d_2}$ possibilities to connect d_2 edges to the k_2 input nodes that have only a non-systematic description at the receiver and $\binom{k-k_2}{d-d_2}$ possibilities to connect the remaining $d - d_2$ edges to the $k - k_2$ received systematic nodes, whereas the total number of possibilities to connect d edges to k nodes is $\binom{k}{d}$. □

In Figure 4.2, the row weight distribution $\Omega_2(\xi)$ is depicted for two exemplary binary LT code ensembles of input size $k = 100$ for $k_2 \in \{1, 2, \ldots, 100\}$. The row weight distribution $\Omega_2(\xi)$ for $k_2 = k = 100$ corresponds to $\Omega(\xi)$ of the unstructured ensemble, which is expurgated in general, i.e. $\Omega_{d=0} = 0$. Note that except for the latter case $\Omega_{2,d_2=0} \neq 0$. This fact is emphasised in Figure 4.2 by the dotted line style.



(a) $k_2 \in \{1, 2, \dots, 15\}$

(b) $k_2 \in \{15, 20, \dots, 100\}$

(c) $k_2 \in \{1, 2, \dots, 10\}$

(d) $k_2 \in \{10, 15, \dots, 100\}$

Figure 4.2: Row weight distributions $\Omega_2(\xi)$ for different values of k_2, the number of erased systematic symbols. The dotted lines are used to highlight the non-zero probability of rows of weight zero.
Code details:
(a) – (b) binary conventionally systematic sparse random ensemble with an input size of $k = 100$ and an average row weight of $\bar{d} = 6$.
(c) – (d) binary conventionally systematic standard random ensemble with an input size of $k = 100$ and an average row weight of $\bar{d} = 50$.

4.2 Bounds on the Conditional Erasure Probabilities

The input symbols can be separated into two sets: the set of k_1 systematically received input nodes and the set of k_2 input symbols that depend on the $k(\gamma_\mathrm{R}-1)+k_2$ received non-systematic encoded symbols. The conditional symbol erasure probability $P^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big)$ is the average of the conditional symbol erasure probabilities $P_1^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big)$ and $P_2^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big)$ of the two sets:

$$P^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big) = \frac{k_1}{k}\cdot P_1^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big) + \frac{k_2}{k}\cdot P_2^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big). \tag{4.7}$$

As the systematically received input nodes are known, i.e. $P_1^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big) = 0$, this results in

$$P^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big) = \frac{k_2}{k}\cdot P_2^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big), \tag{4.8}$$

while the conditional word erasure probability $P^{[\mathfrak{L}]}\big(\not{W}\,\big|k_2\big)$ is simply given by the word erasure probability of the residual ensemble $\mathbf{G}_2$.

$$P^{[\mathfrak{L}]}\big(\not{W}\,\big|k_2\big) = P_2^{[\mathfrak{L}]}\big(\not{W}\,\big|k_2\big), \tag{4.9}$$

So the erasure correction performance of this LT code ensemble is fully determined by the subgraph defined by matrix $\mathbf{G}_2$. Therefore, the bounds on the conditional symbol or word erasure probability can be calculated by applying Theorems 3.14, 3.15, 3.17 and 3.18 to an LT code ensemble that is described by the parameters of matrix $\mathbf{G}_2$, i.e. the row weight distribution $\Omega_2(\xi)$ as in Lemma 4.1 and the dimensions of $\mathbf{G}_2$.

With the previous discussion, the construction of the bounds should be self-evident. Thus, the bounds on the conditional symbol and word erasure probabilities of a conventionally systematic LT code ensemble $\mathfrak{L}$ with generator matrix $\mathbf{G}_{\mathrm{sys,R}} \in \mathbb{F}_q^{n_\mathrm{R}\times k}$ at the receiver as in (4.1) are briefly provided in the following four corollaries without further proof, given that $n_\mathrm{R} = k\gamma_\mathrm{R}$ encoded symbols are received and that k_2 of the k systematic encoded symbols are erased on the SEC.

Corollary 4.2. A lower bound on the conditional symbol erasure probability $P^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big)$ is

$$\underline{P}^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big) = \frac{k_2}{k}\left[\sum_{d_2\in\mathcal{D}_2}\Omega_{d_2}\left(1-\frac{d_2}{k_2}\right)\right]^{k(\gamma_\mathrm{R}-1)+k_2}. \tag{4.10}$$

Corollary 4.3. A lower bound on the conditional word erasure probability $P^{[\mathfrak{L}]}\big(\not{W}\,\big|k_2\big)$ is

$$\underline{P}^{[\mathfrak{L}]}\big(\not{W}\,\big|k_2\big) = \sum_{i=1}^{k}(-1)^{i+1}\binom{k_2}{i}\left(\sum_{d_2\in\mathcal{D}_2}\Omega_{d_2}\frac{\binom{k_2-i}{d_2}}{\binom{k_2}{d_2}}\right)^{k(\gamma_\mathrm{R}-1)+k_2}. \tag{4.11}$$

Corollary 4.4. An upper bound on the conditional symbol erasure probability $P^{[\mathfrak{L}]}\big(\cancel{S}\,\big|k_2\big)$ is

$$\begin{aligned}\overline{P}^{[\mathfrak{L}]}\big(\cancel{S}\,\big|k_2\big) = {} & \frac{k_2}{k}\sum_{w=1}^{k_2-1}\binom{k_2-1}{w-1}(q-1)^{w-1} \\ & \cdot\left[\frac{1}{q}+\frac{q-1}{q}\sum_{d_2\in\mathcal{D}_2}\Omega_{d_2}\cdot\frac{\mathcal{K}_{d_2}(w;k_2)}{\mathcal{K}_{d_2}(0;k_2)}\right]^{k(\gamma_\mathrm{R}-1)+k_2} .\end{aligned} \tag{4.12}$$

Corollary 4.5. An upper bound on the conditional word erasure probability $P^{[\mathfrak{L}]}\big(\cancel{W}\,\big|k_2\big)$ is

$$\begin{aligned}\overline{P}^{[\mathfrak{L}]}\big(\cancel{W}\,\big|k_2\big) = {} & \sum_{w=1}^{k_2}\binom{k_2}{w}(q-1)^{w-1} \\ & \cdot\left[\frac{1}{q}+\frac{q-1}{q}\sum_{d_2\in\mathcal{D}_2}\Omega_{d_2}\cdot\frac{\mathcal{K}_{d_2}(w;k_2)}{\mathcal{K}_{d_2}(0;k_2)}\right]^{k(\gamma_\mathrm{R}-1)+k_2} .\end{aligned} \tag{4.13}$$

The above bounds are evaluated and discussed in Section 4.4 and are depicted in Figures 4.3, 4.4 and 4.5.

4.3 Bounds on the Symbol and Word Erasure Probability

In order to formulate the residual erasure probabilities $P^{[\mathfrak{L}]}\big(\cancel{S}\,\big|\epsilon\big)$ and $P^{[\mathfrak{L}]}\big(\cancel{W}\,\big|\epsilon\big)$ as well as their bounds for a specific channel quality ϵ on the SEC, the probability distribution $P(k_2|\epsilon)$ of the number k_2 of erased systematic encoded symbols depending on the channel erasure probability ϵ is required first. So, assuming a transmission over an SEC with erasure probability ϵ, i.e. on the channel each transmitted encoded symbol is erased independently with a certain probability ϵ, the required probability distribution $P(k_2|\epsilon)$ results in:

$$P(k_2|\epsilon) = \binom{k}{k_2}\epsilon^{k_2}(1-\epsilon)^{k-k_2}. \tag{4.14}$$

Finally, the residual erasure probabilities $P^{[\mathfrak{L}]}\big(\cancel{S}\,\big|\epsilon\big)$ and $P^{[\mathfrak{L}]}\big(\cancel{W}\,\big|\epsilon\big)$ are

$$P^{[\mathfrak{L}]}\big(\cancel{S}\,\big|\epsilon\big) = \sum_{k_2=1}^{k} P(k_2|\epsilon)\cdot P^{[\mathfrak{L}]}\big(\cancel{S}\,\big|k_2\big) \tag{4.15}$$

and

$$P^{[\mathfrak{L}]}\big(\cancel{W}\,\big|\epsilon\big) = \sum_{k_2=1}^{k} P(k_2|\epsilon)\cdot P^{[\mathfrak{L}]}\big(\cancel{W}\,\big|k_2\big). \tag{4.16}$$

However, since closed form expressions of $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{S}}\,\big|k_2\big)$ and $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\,\big|k_2\big)$ are not known in general, only measurements thereof can be used in (4.15) and (4.16). Nevertheless, by inserting the corresponding upper and lower bounds on $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{S}}\,\big|k_2\big)$ and $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\,\big|k_2\big)$ that are given in (4.10) – (4.13) into (4.15) and (4.16), upper and lower bounds on $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{S}}\,\big|\epsilon\big)$ and $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\,\big|\epsilon\big)$ can be easily determined.

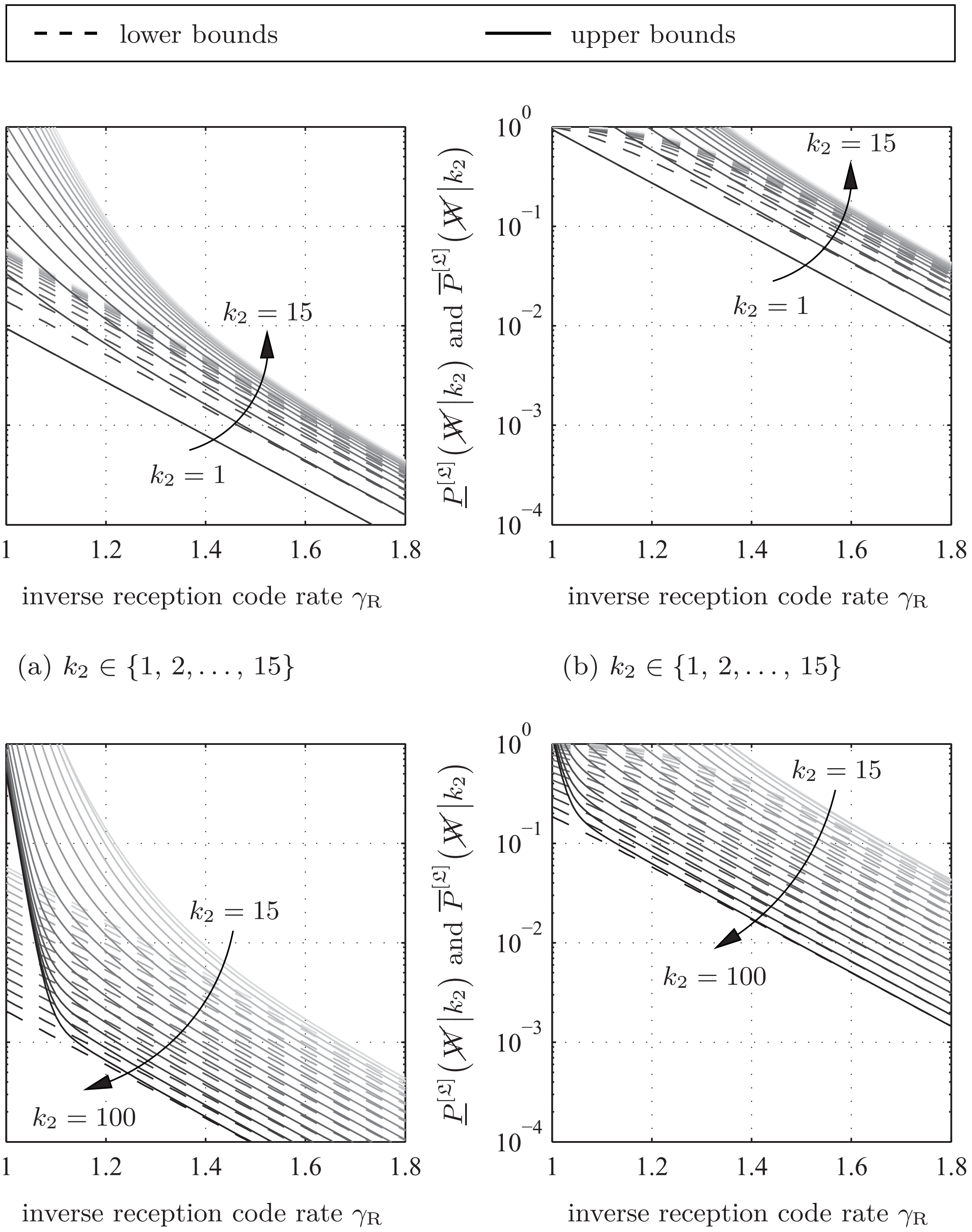
4.4 Numerical Evaluation and Monte Carlo Simulations

In this section, the bounds on the conditional erasure probabilities $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{S}}\,\big|k_2\big)$ and $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\,\big|k_2\big)$ as given in Corollaries 4.2 – 4.5 will be numerically evaluated. However, in order not to restrain the results to the often used SEC, where $P(k_2|\epsilon)$ is a binomial distribution like in (4.14), the numerical evaluation shall at first be limited to the bounds on the conditional probabilities. Hereby, the results are not blurred by assuming a certain channel model. A high value k_2 of erased systematic symbols infers a bad instantaneous channel and a low value for k_2 indicates a good instantaneous channel.

Nevertheless, it is afterwards exemplarily assumed that encoded symbols are indeed independently erased on the channel with equal probability ϵ. For three different values of ϵ representing a good, a medium and a bad channel, i.e. $\epsilon \in \{0.01, 0.1, 0.5\}$, the residual erasure probabilities (4.15) and (4.16) are determined by means of Monte Carlo simulations and the corresponding bounds are computed.

In Figure 4.3 upper and lower bounds on $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{S}}\,\big|k_2\big)$ and $P^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\,\big|k_2\big)$ are depicted as a function of the inverse reception code rate γ_{R} for a binary sparse random ensemble of input size $k = 100$ and density $\Delta = 0.06$. The left subfigures contain the bounds on symbol level and the right subfigures the ones on word level. As the residual erasure probability is not monotonic in k_2 it has been split up into two monotonic parts. In the upper two subfigures the residual erasure probability rises with k_2 up to $k_2 \approx 15$. But with a degrading instantaneous channel, i.e. for increasing k_2, the residual erasure probability monotonically decreases again until it reaches the level of the unstructured ensemble with $k_2 = 100$, which constitutes the minimal residual erasure probability. This is depicted in the lower two subfigures.

Taking a look at the bounds from a different perspective in Figures 4.5(a) and 4.5(b) reveals very clearly that for relevant values of γ_{R} the absolute minimum is attained at $k_2 = 100$, i.e. for the ensemble without systematic prefix. This outcome corresponds to the claims in the literature [SL05, Sho06, SL11]. Although this is only an example, the result can be transferred to other ensembles with (not too) sparse generator matrices, not just the sparse random ensembles. So for such ensembles, conventionally systematic encoding is not advisable, except



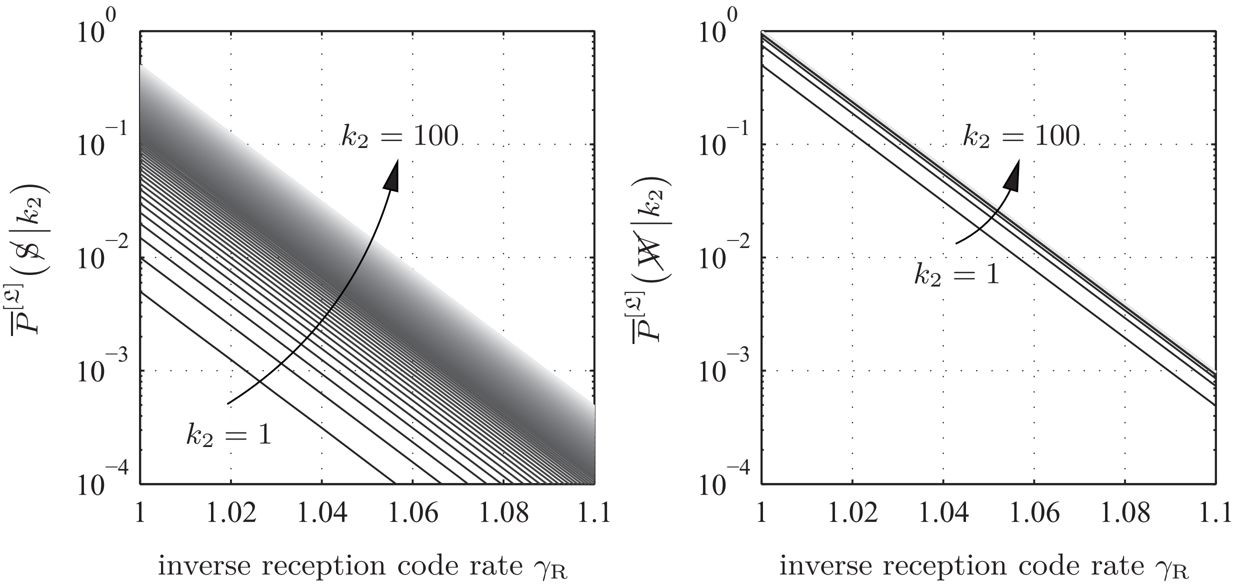
(a) $k_2 \in \{1, 2, \ldots, 15\}$

(b) $k_2 \in \{1, 2, \ldots, 15\}$

(c) $k_2 \in \{15, 20, \ldots, 100\}$

(d) $k_2 \in \{15, 20, \ldots, 100\}$

Figure 4.3: Upper and lower bounds (4.10) – (4.13) on the residual erasure probability on symbol (left) and word level (right) conditioned on the number k_2 of erased systematic symbols as a function of the inverse reception code rate γ_{R}.
Code details: binary conventionally systematic sparse random ensemble with an input size of $k = 100$ and an average row weight of $\bar{d} = 6$. As the dependence on k_2 is non-monotonous, the bounds are arranged in different subfigures for low and high values of k_2, respectively.



(a) $k_2 \in \{1, 2, \ldots, 100\}$

(b) $k_2 \in \{1, 2, \ldots, 100\}$

Figure 4.4: Upper bounds (4.12) and (4.13) on the residual erasure probability on symbol and word level conditioned on the number k_2 of erased systematic symbols as a function of the inverse reception code rate γ_{R}.
Code details: binary conventionally systematic standard random ensemble with input size $k = 100$ and consequently an average row weight of $\bar{d} = 50$.

for extremely good channels where the probability $P(k_2 = 0|\epsilon)$ is sufficiently large, such that the systematic part survives as a whole most of the time.

The non-monotonic behaviour will be illustrated by means of some examples. A necessary condition for an input symbol to be recovered if it has not been received in the conventionally systematic part, is to be covered by $\mathbf{G}_2$. This means that in the subgraph given by $\mathbf{G}_2$ the respective input symbol is connected to an encoded symbol. The probability that this necessary condition is not met is given by the lower bound $\underline{P}^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big)$ in (4.10). To simplify the following explanation it shall be assumed that $\tilde{\mathbf{G}}_{\mathrm{sys,R}}$ is a square matrix, i.e. $\gamma_{\mathrm{R}} = 1$. If $k_2 = 1$, $\mathbf{G}_2$ is a scalar and takes a non-zero value with probability 0.06 which corresponds to the density of the current ensemble and $\underline{P}^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2 = 1\big) = \frac{k_2}{k}(1-\Delta)^{k_2} = \frac{1}{100} \cdot 0.94 = 0.0094$. For low but increasing values of k_2, the growing fraction $\frac{k_2}{k}$ of unrecovered symbols dominates $\underline{P}^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2\big)$ until for medium values of k_2 the falling exponential term $(1-\Delta)^{k_2}$ starts to take over, so that $\underline{P}^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2 = 2\big) \approx 0.0177$ and $\underline{P}^{[\mathfrak{L}]}\big(\not{S}\,\big|k_2 = k\big) \approx 0.002055$.

Moreover, for low values of k_2, most of the initial systematically unrecovered input symbols are also recovered systematically but from the weight-one rows of the matrix $\mathbf{G}_2$. Note that the probability of an all-zero row in $\mathbf{G}_2$ is not zero and particularly for low k_2 it is not even small as can be observed in Figures 4.2(a) and

4.2(b) on page 82. For medium values of k_2 an increasing number of rows in $\mathbf{G}_2$ has two or more entries, but the resulting code is too weak to recover many unknowns. As k_2 increases further, the code gets stronger and although the probability of row weight $d_2 = 1$ decreases dramatically, i.e. systematic decoding from $\mathbf{G}_2$ gets less likely, the coding gain for higher k_2 outweighs the benefit of systematic decoding from $\mathbf{G}_2$ for low k_2.

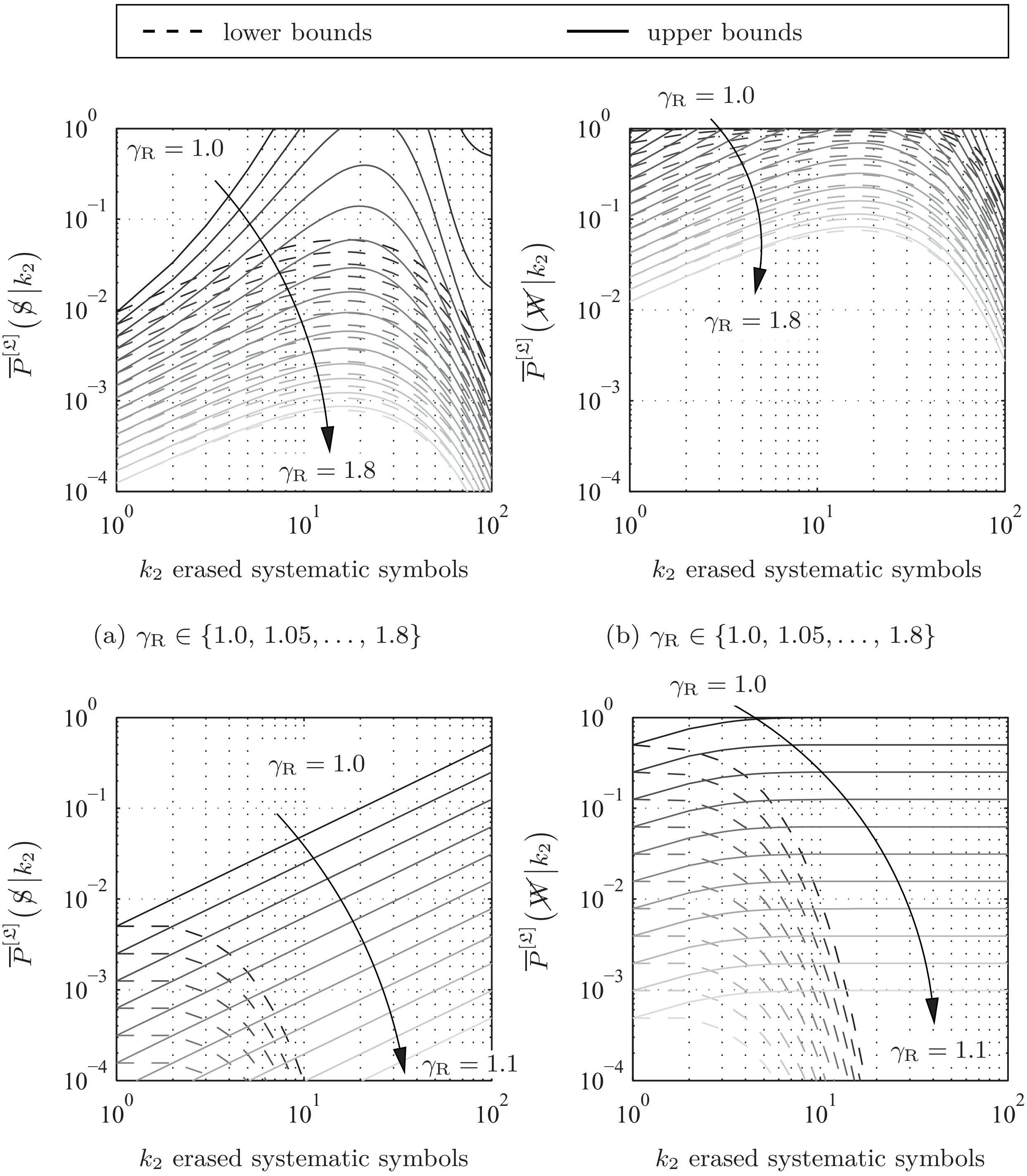
Although a conventionally systematic prefix is not useful in the just given example ensemble, it can be for others, such as the standard random ensemble. In Figure 4.4 the upper bounds on the conditional erasure probability are given for $k_2 \in \{1, 2, \ldots, 100\}$ as a function of the inverse reception code rate γ_{R}. Also, the upper and lower bounds are depicted in Figure 4.5(c) and 4.5(d) as a function of k_2 and parametrised with γ_{R}.

In contrast to the sparse random ensemble, the upper bounds of the standard random ensemble are monotonically increasing as a function of k_2. As the upper bounds are again very close to the true values, the upper bounds are used instead to assess the performance of this ensemble. The lower bounds diverge quickly from their respective upper bounds and play only a minor role, since usually rank deficiencies in standard random ensembles do not arise from unconnected input nodes. Thus, the lower bounds of standard random ensembles are mostly omitted.

While the upper bound on symbol level is linear in k_2, the one on word level remains constant over a wide range of k_2. The explanation for the different behaviour of the standard random ensemble lies in the quasi-independence of the erasure correction performance from the input size. Its performance depends almost solely on the absolute symbol reception overhead and on the field order, as has already been observed in Section 3.4. So, when comparing the non-systematic standard random ensemble $\mathbf{G}_{\mathrm{R}} \in \mathbb{F}_2^{n_{\mathrm{R}} \times k}$, where $n_{\mathrm{R}} = k + \eta_{\mathrm{R}}$, with that of the smaller ensemble $\mathbf{G}_2 \in \mathbb{F}_2^{(k_2+\eta_{\mathrm{R}}) \times k_2}$, their performance on word level is almost equal as long as k_2 is greater than 4 or 5, since for too small input sizes the independence assumption becomes invalid. The linear dependence of $P^{[\mathfrak{L}]}\left(\not{S} \middle| k_2\right)$ on k_2 is due to the averaging over systematically and non-systematically received input symbols.

In Figure 4.6, the two ensembles are also numerically evaluated for different symbol erasure probabilities of the channel, i.e. $\epsilon \in \{0.01, 0.1, 0.5\}$. Together with the upper and lower bounds, also the simulated residual erasure rates are depicted as well as the erasure rates of the respective unstructured ensemble. The latter ones serve as a reference and allow to easily check whether or not a conventionally systematic prefix is beneficial.

On the left, the performance of the sparse random ensemble is depicted. It is interesting to see that for good channels the simulated curves rather tend towards the lower bound. However, this is still insufficient to approach the performance of the unstructured sparse random ensemble for neither one of the channel qualities.



(a) $\gamma_{\mathrm{R}} \in \{1.0, 1.05, \ldots, 1.8\}$

(b) $\gamma_{\mathrm{R}} \in \{1.0, 1.05, \ldots, 1.8\}$

(c) $\gamma_{\mathrm{R}} \in \{1.0, 1.01, \ldots, 1.1\}$

(d) $\gamma_{\mathrm{R}} \in \{1.0, 1.01, \ldots, 1.1\}$

Figure 4.5: Upper and lower bounds (4.10) – (4.13) on the residual erasure probability on symbol and word level conditioned on the number k_2 of erased systematic symbols for different values of the inverse reception code rate γ_{R}.
Code details:
(a) – (b) binary conventionally systematic sparse random ensemble with an input size of $k = 100$ and an average row weight of $\bar{d} = 6$.
(c) – (d) binary conventionally systematic standard random ensemble with an input size of $k = 100$ and an average row weight of $\bar{d} = 50$.

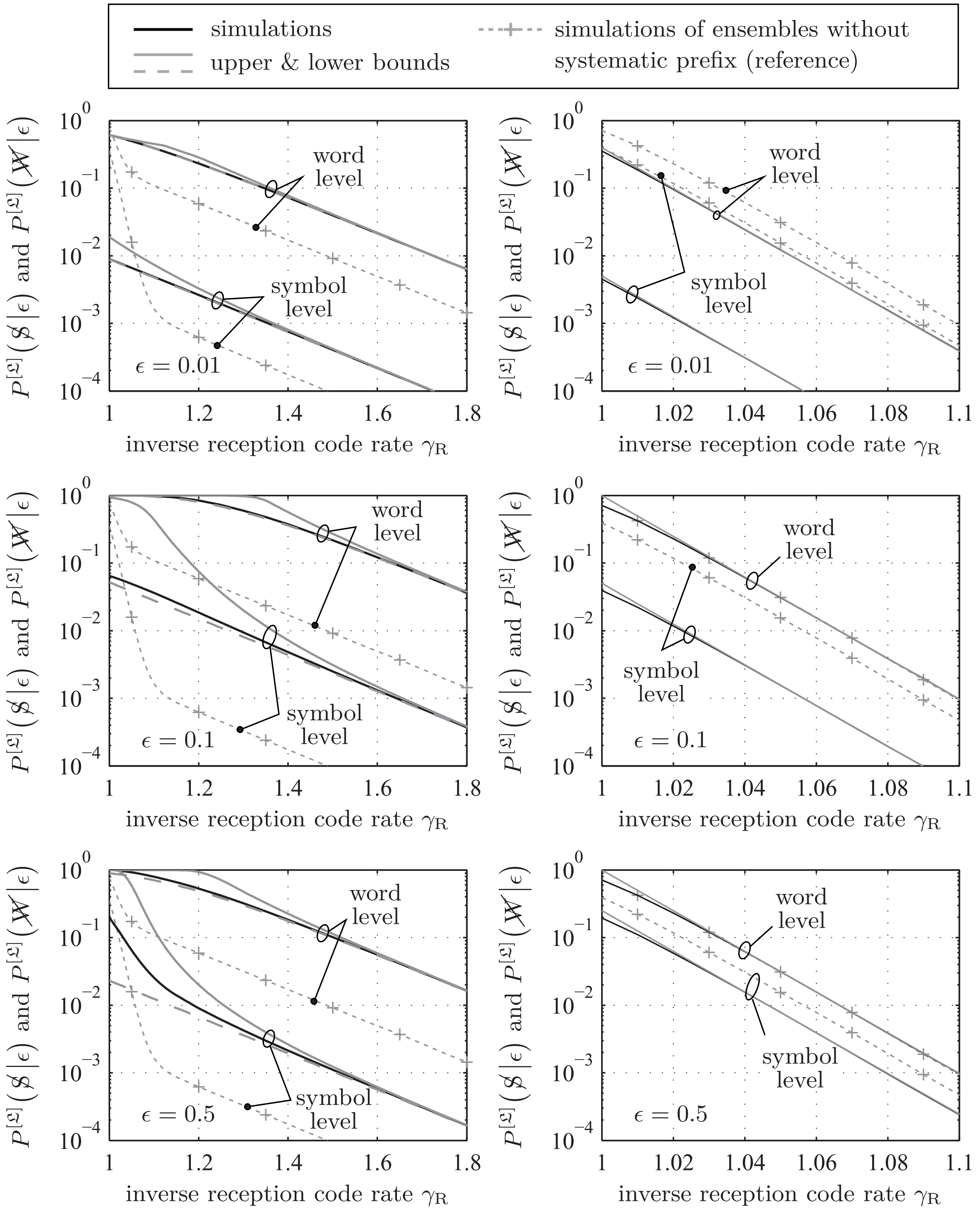


Figure 4.6: Upper (and lower) bounds on the residual erasure probability on word and symbol level as well as the respective simulated erasure rates, all conditioned on the erasure probability $\epsilon \in \{0.01,\, 0.1,\, 0.5\}$ of the channel. Code details: two binary ensembles with $k = 100$ and a conventionally systematic prefix.
– Left figures: sparse random ensemble with $\Delta = 0.06$
– Right figures: standard random ensemble with $\Delta = 0.5$, no lower bounds.
The dotted lines with crosses mark the respective ensembles without prefix.

On the right, the standard random ensemble has been evaluated. As the lower bounds are of minor importance in this case, they have been omitted. For a very good channel, i.e. with $\epsilon = 0.01$, a small performance gain on word level and a significant gain on symbol level can be observed compared to the unstructured ensemble. Yet, for a somewhat worse channel, i.e. with $\epsilon = 0.1$, the performance gain on word level becomes essentially zero, though on symbol level it is still considerable. In a bad channel, e.g. with $\epsilon = 0.5$, the gain on symbol level also decreases to a rather small amount. The possible gain from a conventionally systematic prefix is therefore much larger on symbol level than on word level. This implies that the conventionally systematic prefix induces a shift of high-weight residual erasures (i.e. many residual symbol erasures in an information word) to low-weight erasures. A received yet still unrecoverable information word contains therefore fewer erasures on average than in case of an unstructured ensemble, which is particularly advantageous if high-rate precoding is intended.

These results show that for the standard random ensemble it is profitable to employ a conventionally systematic prefix. This contradicts the one-sided view from the literature, e.g. [SL11]. Not only does it yield a better erasure correction performance or at least one that is as good as that of the unstructured ensemble, but also the complexity is reduced from $\mathcal{O}(k^3)$ to $\mathcal{O}(k_2^3)$, which makes a tremendous difference particularly in good channels, where usually $k_2 \ll k$.

Chapter 5

Precodes

To enhance the erasure correction performance of LT code ensembles, most of which naturally suffer from an erasure floor, one or more precode stages can be used to lower the erasure floor. This concatenation of a rateless code with a precode is denoted Raptor code [Sho04,Sho06] and has already been briefly addressed in Section 2.3. In this chapter, the erasure correction properties of different types of fixed-rate codes will be determined and discussed under optimal erasure decoding. Then, in the next chapter, the erasure correction properties of some suitable precodes and of LT code ensembles are combined in order to accurately determine the erasure correction properties of Raptor code ensembles.

One important requirement for a code to qualify as a precode is a very high code rate. Moreover, a good precode should reliably correct low-weight erasure patterns, i.e. it should have a Hamming weight distribution with a low probability of occurrence of low-weight codewords. In the following, the erasure correction properties of different high-rate codes are provided in terms of the decoding failure probability $P\big(\cancel{\mathcal{W}}\,\big|\,\mathsf{e}=e\big)$ given e erasures in the received codeword. The notation in this chapter is simplified and disregards the existence of a rateless component, i.e. unlike the notation introduced in Section 2.3, the length of the information vector is k, the codeword length is n with $m = n - k$, the number of parity symbols, and the code rate is $\rho = k/n$. Since only block codes are considered in the following, the codes are specified by means of the characterising triple (n, k, $\delta_{\mathrm{H,min}}$) for binary codes and the quadruple (n, k, $\delta_{\mathrm{H,min}}$, q) in case of non-binary codes.

5.1 Deterministic Precodes

In this section, the erasure correction properties of various codes with a deterministic code construction will be provided. Starting with the ideal performance of general maximum distance separable (MDS) codes, the decoding failure probability of shortened non-binary Hamming codes is determined as well as that of shortened extended Hamming codes.

5.1.1 Maximum Distance Separable Codes

Maximum distance separable (MDS) codes, e.g. Reed-Solomon codes, have a minimum Hamming distance of $\delta_{\mathrm{H,min}} = n - k + 1$ and are able to correct e erasures in the received codeword if $e \leq m$, where $m = n - k$. So the decoding failure probability of this type of code is simply

$$P\big(\cancel{W}\,\big|\,\mathsf{e} = e\big) = \begin{cases} 0 & \text{if } 0 < e \leq m \\ 1 & \text{if } e > m. \end{cases} \tag{5.1}$$

5.1.2 Hamming Codes over $\mathbb{F}_q$

The characterising triple of an ordinary binary Hamming code $\mathfrak{H}$ with k input bits, a codeword length of n and m parity bits is provided by

$$(n,\, k,\, \delta_{\mathrm{H,min}}) = (2^m - 1,\, 2^m - m - 1,\, 3). \tag{5.2}$$

Hamming codes can be generalised straightforward to any field order q [Ulr57], which results in the characteristic quadruple

$$(n,\, k,\, \delta_{\mathrm{H,min}},\, q) = \left(\frac{q^m - 1}{q - 1},\, \frac{q^m - 1}{q - 1} - m,\, 3,\, q\right). \tag{5.3}$$

Alike, by shortening Hamming codes by σ bits or symbols, i.e. by fixing σ data bits or symbols to be zero and omitting their transmission, in order to adapt them to any convenient input size k, the previous quadruple becomes

$$(n - \sigma,\, k - \sigma,\, \delta_{\mathrm{H,min}},\, q) = \left(\frac{q^m - 1}{q - 1} - \sigma,\, \frac{q^m - 1}{q - 1} - m - \sigma,\, 3,\, q\right), \tag{5.4}$$

where

$$\begin{aligned} & 0 \leq \sigma < \left(\frac{q^m - 1}{q - 1} - m\right) - \left(\frac{q^{m-1} - 1}{q - 1} - (m - 1)\right) \\ \Longrightarrow\quad & 0 \leq \sigma < q^{m-1} - 1. \end{aligned} \tag{5.5}$$

Notwithstanding the long known fact that Hamming codes can correct more erasures than $\delta_{\mathrm{H,min}} - 1$, a thorough analysis of the erasure correction probability could not be found in the respective literature. However, with the help of Lemma 1 in [ZLJR08] and the herein after made generalisations thereof, it is possible to derive the exact erasure correction probability for a certain number of erasures within a received codeword, be it encoded by a binary or a non-binary Hamming code, shortened or unshortened. Note that here the shortening positions are chosen uniformly at random and thus the exact erasure correction probability is the average over all shortened code realisations of a particular parametrisation. An optimised choice of the shortening positions undoubtedly yields slightly better results.

Before approaching Lemma 1 from [ZLJR08], some prerequisites shall be discussed beforehand, mostly to introduce the used notation. As the shortened q-ary Hamming code $\mathfrak{H}_{q,\sigma}$ is the most general one, for either σ can be set to zero to obtain an unshortened code or q can be set to any prime power, it shall be exclusively used in the following.

Optimal erasure decoding of a shortened q-ary Hamming code, which possesses a q-ary parity-check matrix $\mathbf{H}$ of size $m \times (n - \sigma)$, is commonly performed by means of syndrome erasure decoding. By design and definition $\mathbf{Hy} = \mathbf{0}$, where $\mathbf{y}$ is a codeword. Let e erasures occur on the channel, then $\mathbf{e}$ denotes a vector of length e containing the positions at which the received word contains erasures. If the code is used as a precode to a rateless code, the number of erasures e corresponds to the erasure weight w_e of the intermediate codeword. Now, let $\mathbf{y_e}$ denote those e symbols of the codeword that have been erased and let $\mathbf{y_{\bar{e}}}$ be the symbols that have been received. Accordingly, let $\mathbf{H_e}$ contain those e columns of $\mathbf{H}$ that are associated with the erased symbols and let $\mathbf{H_{\bar{e}}}$ contain the $\bar{e} = n - \sigma - e$ ones associated with the non-erased symbols, then

$$\underbrace{\mathbf{H_e y_e}}_{\mathbf{s}} + \underbrace{\mathbf{H_{\bar{e}} y_{\bar{e}}}}_{-\mathbf{s}} = \mathbf{0}, \tag{5.6}$$

where $\mathbf{s}$ is known as the syndrome vector. In (5.6) all quantities but $\mathbf{y_e}$ are known at the receiver. To allow a recovery of $\mathbf{y_e}$ and consequently of $\mathbf{y}$ as a whole by solving the corresponding system of linear equations $\mathbf{H_e y_e} = \mathbf{s}$, the matrix $\mathbf{H_e}$ is required to have full column rank, i.e. $\mathrm{rank}(\mathbf{H_e}) = e$.

Given a certain number e of erased symbols, their position is randomly determined by the symbol erasure channel and such is the choice of the columns from $\mathbf{H}$ that compose $\mathbf{H_e}$. So the successful decoding probability equals the probability of randomly sampling a full rank matrix $\mathbf{H_e}$ from $\mathbf{H}$. The number of full rank matrices $\mathbf{H_e}$ of size $m \times e$ is given by the already mentioned Lemma 1 from [ZLJR08]. As a shortened q-ary Hamming code $\mathfrak{H}_{q,\sigma}$ is considered here, the corresponding lemma and its proof will be generalised to this case based on the original one for the unshortened binary case.

Lemma 5.1 (Generalisation of Lemma 1 in [ZLJR08])**.** Let $\mathbf{H_e}$ be a q-ary matrix of size $m \times e$ whose columns are equal to e columns of a shortened q-ary Hamming code's parity-check matrix $\mathbf{H}$ of size $m \times (n - \sigma)$. Then the number of matrices $\mathbf{H_e}$ that have full column rank, i.e. $\mathrm{rank}(\mathbf{H_e}) = e$, equals

$$N^{[\mathfrak{H}_{q,\sigma}]}(e,\, m,\, \sigma) = \begin{cases} \frac{1}{e!} \prod\limits_{i=0}^{e-1} \left(\frac{q^m - q^i}{q-1} - \sigma \right) & \text{if } 0 < e \leq m \\ 0 & \text{if } e > m. \end{cases} \tag{5.7}$$

Proof. The columns of the parity-check matrix $\mathbf{H}$ of length $n - \sigma = \frac{q^m-1}{q-1} - \sigma$ are all $\frac{q^m-1}{q-1}$ non-zero q-ary m-tuples that span the m-dimensional space minus the σ removed ones due to shortening. Consequently, the number of matrices $\mathbf{H_e}$ that have full column rank, i.e. $\mathrm{rank}(\mathbf{H_e}) = e$, is equal to the number of different bases of e-dimensional subspaces of the m-dimensional space. Let $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_e\}$ denote the set of basis vectors of an e-dimensional subspace. Then, the number of such sets can be determined as follows:

- There are $n - \sigma = \frac{q^m-1}{q-1} - \sigma$ possibilities to select the first vector $\mathbf{b}_1$ from the non-zero m-tuples.
- The second non-zero vector $\mathbf{b}_2$ is chosen such that $\mathbf{b}_2 \neq c_1\mathbf{b}_1$ with $c_1 \in \mathbb{F}_q \setminus \{0\}$. Since by construction, none of the $q-1$ multiples of any column are contained in $\mathbf{H}$, the condition reduces to $\mathbf{b}_2 \neq \mathbf{b}_1$ when $c_1 = 1$, i.e. there are $\frac{q^m-1}{q-1} - \frac{q-1}{q-1} - \sigma = \frac{q^m-1}{q-1} - 1 - \sigma = n - 1 - \sigma$ choices for $\mathbf{b}_2$.
- In general, for $i = 3, 4, \ldots, e$ the non-zero base vectors $\mathbf{b}_i$ are chosen such that they differ from any linear combination of the $i-1$ previously determined base vectors, i.e. $\mathbf{b}_i \neq \sum_{j=1}^{i-1} c_j\mathbf{b}_j$ with $c_j \in \mathbb{F}_q$. For picking a non-zero vector *within* the span of the $i-1$ previous base vectors, there are $q^{i-1} - 1$ choices of $c_j \in \mathbb{F}_q$ with $1 \leq j < i$ and $(c_1 = c_2 = \ldots = c_{i-1})^{\mathsf{T}} \neq \mathbf{0}^{\mathsf{T}}$. The exclusion of multiples results in $\frac{q^{i-1}-1}{q-1}$ choices. *Avoiding* this span results in $n - \frac{q^{i-1}-1}{q-1} - \sigma = \frac{q^m-q^{i-1}}{q-1} - \sigma$ choices for $\mathbf{b}_i$.
- The final number $N^{[\mathfrak{H}_{q,\sigma}]}(e, m, \sigma)$ of matrices $\mathbf{H_e}$ is thus obtained as the product of the number of all just mentioned possibilities to determine the respective base vectors without taking their order into account. □

Theorem 5.2. The failure probability of optimal erasure decoding of a received sequence of $n-\sigma = \frac{q^m-1}{q-1} - \sigma$ symbols that contains e erasures at random positions, where the sequence has been encoded using a shortened q-ary Hamming code $\mathfrak{H}_{q,\sigma}$, is given by

$$P^{[\mathfrak{H}_{q,\sigma}]}\left(\mathcal{W} \middle| \mathsf{e} = e\right) = \begin{cases} 1 - \frac{(n-\sigma-e)!}{(n-\sigma)!} \prod_{i=0}^{e-1} \left(\frac{q^m-q^i}{q-1} - \sigma\right) & \text{if } 0 < e \leq m \\ 1 & \text{if } e > m. \end{cases} \tag{5.8}$$

Proof. The successful decoding probability can be expressed as the number $N^{[\mathfrak{H}_{q,\sigma}]}(e, m, \sigma)$ of full rank matrices $\mathbf{H_e}$, provided in Lemma 5.1, divided by the number of all possible matrices $\mathbf{H_e}$

$$P^{[\mathfrak{H}_{q,\sigma}]}\left(\mathcal{W} \middle| \mathsf{e} = e\right) = 1 - \frac{N^{[\mathfrak{H}_{q,\sigma}]}(e, m, \sigma)}{\binom{n-\sigma}{e}}. \tag{5.9}$$

The denominator represents the number of possibilities to choose e columns without repetition from the $n - \sigma$ columns of the Hamming code's $\mathfrak{H}_{q,\sigma}$ parity-check matrix $\mathbf{H}$. □

5.1.3 Extended Hamming Codes

Extended Hamming codes are derived from ordinary binary Hamming codes by adding a further parity check equation over all bits. Particularly for erasure decoding, this increase of the codeword length by one bit is advantageous as it directly increases $\delta_{\mathrm{H,min}}$ from three to four. Unfortunately, this extension by means of an additional parity check equation cannot be generalised in a meaningful way to a non-binary domain. Attempting to do so, leads to a code rate that quickly approaches zero as the codeword length increases, rendering non-binary extended Hamming codes rather useless.

An extended binary Hamming code $\mathfrak{H}^+$ with k input bits, a codeword length of n and $m = m' + 1$ parity bits is characterised by the triple

$$(n,\, k,\, \delta_{\mathrm{H,min}}) = \left(2^{m'},\, 2^{m'} - m',\, 4\right), \tag{5.10}$$

where m' here denotes the number of parity bits of the underlying Hamming code. Also extended Hamming codes can be adapted to any suitable input size by shortening them by σ bits, such that the more general characterising triple becomes

$$(n - \sigma,\, k - \sigma,\, \delta_{\mathrm{H,min}}) = \left(2^{m'} - \sigma,\, 2^{m'} - m' - \sigma,\, 4\right), \tag{5.11}$$

where

$$\begin{aligned} & 0 \leq \sigma < \left(2^{m'} - m'\right) - \left(2^{m'-1} - \left(m' - 1\right)\right) \\ \Longrightarrow \quad & 0 \leq \sigma < 2^{m'-1} - 1. \end{aligned} \tag{5.12}$$

As in the case of Hamming codes, an expression for the erasure correction capability of extended Hamming codes beyond $\delta_{\mathrm{H,min}} - 1$ is not to be found in common literature. Therefore, following the previous rationale, the required erasure correction probability will be briefly laid out subsequently. In order to derive the failure probability of shortened extended Hamming codes under optimal erasure decoding, Lemma 1 from [ZLJR08] or Lemma 5.1, respectively, needs to be adapted to the current type of codes. The proof of the adaptation follows along similar lines as above.

Lemma 5.3. Let $\mathbf{H_e}$ be a binary matrix of size $(m' + 1) \times e$ whose columns are equal to e columns of a shortened extended Hamming code's parity-check matrix

$\mathbf{H}$ of size $(m'+1)\times(n-\sigma)$. Then the number of matrices $\mathbf{H_e}$ that have full column rank, i.e. $\mathrm{rank}(\mathbf{H_e}) = e$, equals to

$$N^{[\mathfrak{H}_\sigma^+]}\big(e,\, m',\, \sigma\big) = \begin{cases} \frac{2^{m'}-\sigma}{e!} \prod\limits_{i=0}^{e-2} \Big(2^{m'} - 2^i - \sigma\Big) & \text{if } 0 < e \le m'+1 \\ 0 & \text{if } e > m'+1. \end{cases} \tag{5.13}$$

Proof. In its systematic form the parity-check matrix $\mathbf{H}$, which is of size $(m'+1)\times(n-\sigma)$, consists of all except σ odd weight $(m'+1)$-tuples, i.e. its $2^{m'}-\sigma$ columns are constituted by all odd-weight $(m'+1)$-tuples minus the σ removed ones due to shortening. And the number of matrices $\mathbf{H_e}$ that have full column rank, i.e. $\mathrm{rank}(\mathbf{H_e}) = e$, is equal to the number of different bases of e-dimensional subspaces of the $(m'+1)$-dimensional space. Let $\{\mathbf{b}_1,\, \mathbf{b}_2, \ldots,\, \mathbf{b}_e\}$ denote the set of base vectors of an e-dimensional subspace. Then, the number of such sets can be determined as follows:

- There are $n-\sigma = 2^{m'}-\sigma$ possibilities to select the first vector $\mathbf{b}_1$ from the non-zero $(m'+1)$-tuples of odd weight, since $\sum_{i\ \mathrm{odd}}\binom{m'+1}{i} = \sum_{i\ \mathrm{even}}\binom{m'+1}{i} = 2^{m'}$.

- The second non-zero vector $\mathbf{b}_2$ is chosen such that $\mathbf{b}_2 \neq \mathbf{b}_1$, for which there are $n-1-\sigma$ choices.

- In general, for $i = 3,\, 4, \ldots,\, e$ the non-zero base vectors $\mathbf{b}_i$ are chosen such that they differ from any linear combination of the $i-1$ previously determined base vectors, i.e. $\mathbf{b}_i \neq \sum_{j=1}^{i-1} c_j \mathbf{b}_j$ with $c_j \in \{0,\, 1\}$. Since the sum of an even number of odd-weight binary vectors yields a vector of even weight, only linear combinations of odd numbers of binary odd-weight vectors need to be considered. So, only those linear combinations need to be avoided where $w = \|(c_1,\, c_2, \ldots,\, c_{i-1})^{\mathsf{T}}\|_{\mathrm{H}}$ is odd, of which there are $\sum_{w\ \mathrm{odd}}\binom{i-1}{w} = 2^{i-2}$ choices. Consequently, there are $n - 2^{i-2} - \sigma$ possibilities to choose $\mathbf{b}_i$.

- Finally, the number $N^{[\mathfrak{H}_\sigma^+]}(e,\, m',\, \sigma)$ of matrices $\mathbf{H_e}$ results as the product of the number of all just mentioned possibilities to determine the respective base vectors, again without taking their order into account. □

Theorem 5.4. The failure probability of optimal erasure decoding of a received sequence of $n-\sigma = 2^{m'}-\sigma$ symbols that contains e erasures at random positions, where the sequence has been encoded using a shortened extended Hamming code

$\mathfrak{H}_\sigma^+$, is given by

$$P^{[\mathfrak{H}_\sigma^+]}\left(\cancel{\mathcal{W}}\middle|\mathsf{e}=e\right) = \begin{cases} 1 - \frac{(n-\sigma-e)!(2^{m'}-\sigma)}{(n-\sigma)!} \prod\limits_{i=0}^{e-2} \left(2^{m'} - 2^i - \sigma\right) & \text{if } 0 < e \leq m'+1 \\ 1 & \text{if } e > m'+1. \end{cases} \tag{5.14}$$

Proof. The successful decoding probability is given as the number $N^{[\mathfrak{H}_\sigma^+]}(e,\, m',\, \sigma)$ of full rank matrices $\mathbf{H}_\mathbf{e}$, which is equal to (5.13) from Lemma 5.3, divided by the number of all possible matrices $\mathbf{H}_\mathbf{e}$

$$P^{[\mathfrak{H}_\sigma^+]}\left(\cancel{\mathcal{W}}\middle|\mathsf{e}=e\right) = 1 - \frac{N^{[\mathfrak{H}_\sigma^+]}(e,\, m',\, \sigma)}{\binom{n-\sigma}{e}}. \tag{5.15}$$

The denominator denotes the number of possibilities to choose e columns without repetition from the $n-\sigma$ columns of the extended Hamming code's $\mathfrak{H}_\sigma^+$ parity-check matrix $\mathbf{H}$. □

5.2 Stochastic Precodes

Apart from the previously discussed deterministic precodes, there exist further linear codes of stochastic nature that appear suited as precodes. The ones that will be dealt with in this section fall into the category of parity-check ensembles $\mathfrak{C}$. These ensembles are mostly defined by means of their parity-check matrix $\mathbf{H}$ of size $m \times n$ which is characterised by two distributions, i.e. the so-called variable node degree distribution $\Lambda(\xi) = \sum_{d=2}^{d_{\mathrm{v,max}}} \Lambda_d \xi^d$ and the check node degree distribution $R(\xi) = \sum_{d=2}^{d_{\mathrm{c,max}}} R_d \xi^d$ both from a node perspective. The coefficients Λ_d or R_d of ξ^d in $\Lambda(\xi)$ or $R(\xi)$ denote the fraction of rows or columns of weight d, respectively.

Due to the concentration effect, the performance of an individual parity-check code $\mathbf{H}$ from an ensemble $\mathfrak{C} = \left(\Pr\{\boldsymbol{\mathsf{H}} = \mathbf{H}\},\, \mathbb{F}_q^{m\times n}\right)$ concentrates around the ensemble average with high probability [RU08]. Nevertheless, for an actual precode implementation, a fixed code instance should be preferably used, particularly one that is carefully chosen from such an ensemble. It should be kept in mind though, that for stochastic parity-check codes the probability of occurrence of low-weight codewords is relatively high, i.e. such codes do not have a good Hamming weight distribution. So these codes exhibit an erasure floor, too, and thus they should only be employed as an intermediate precode if for instance a Hamming-type precode alone is too weak and a precode of a lower code rate is required. The outermost precode should always be a code with a good Hamming weight distribution.

In [RVF07] binary Raptor codes employing precodes taken from a particular kind of LDPC code ensemble have been analysed under ML decoding. The used LDPC precode ensemble is constructed randomly, similarly to the sparse random LT code ensemble, i.e. by setting each entry in a parity-check matrix $\mathbf{H}$ of size $m \times n$ to one or zero according to the outcome of i.i.d. Bernoulli trials parametrised with the probability of sampling a one. Due to the construction method, this ensemble is referred to as the sparse random parity-check ensemble, which can be generalised to higher order Galois fields in a straightforward manner.

In [DPT$^+$02], upper bounds on the residual symbol and word erasure probability of regular LDPC code ensembles as well as of the standard random parity-check ensemble after ML decoding have been derived as a function of the erasure probability on the BEC. Based on this work, the bounds on the word erasure probability have been extended to q-ary ensembles in [LPC13]. Therein the upper bound has also been given in general form for arbitrary degree distributions.

However, this general form depends on the so-called average weight enumerating function (WEF) of an ensemble. And unfortunately, this function is rather difficult or complex to determine for most ensembles other than a few very specific ones like for instance the just mentioned sparse random and standard random ensembles or regular ensembles. For approximating the average WEF in the general case, there exists a method called Hayman approximation or Hayman's method [Wil94], which has first been applied to this problem in [Di04, DRU06] and has been extended to the non-binary case in [KPD$^+$08, KPDS11].

5.2.1 Upper Bounds on Conditional Residual Erasure Probabilities

The upper bounds on word and/or on symbol level in the aforementioned works [DPT$^+$02, RVF07, LPC13] are all based on the same rationale. The upper bound $\overline{P}^{[\mathfrak{C}]}\left(\cancel{W}\middle|\mathsf{e}=e\right)$ on the residual word erasure probability $P^{[\mathfrak{C}]}\left(\cancel{W}\middle|\mathsf{e}=e\right)$ of a parity-check ensemble $\mathfrak{C}$ given e symbol erasures in the encoded sequence, i.e. on the probability $\Pr\{\mathrm{rank}(\boldsymbol{\mathsf{H}}_{\boldsymbol{\mathsf{e}}}) < e\}$ of rank deficiency in the submatrix $\boldsymbol{\mathsf{H}}_{\boldsymbol{\mathsf{e}}}$ is assessed by a union bounding argument, where $\boldsymbol{\mathsf{H}}_{\boldsymbol{\mathsf{e}}}$ is obtained from the original parity-check matrix $\boldsymbol{\mathsf{H}}$ of size $(m \times n)$ by the same means as in the previous case of Hamming codes:

$$
\begin{align}
P^{[\mathfrak{C}]}\left(\cancel{W}\middle|\mathsf{e}=e\right) &= \Pr\{\mathrm{rank}(\boldsymbol{\mathsf{H}}_{\boldsymbol{\mathsf{e}}}) < e\} \tag{5.16}\\
&= \Pr\{\exists \mathbf{y}_{\boldsymbol{\mathsf{e}}} \in \ker(\boldsymbol{\mathsf{H}}_{\boldsymbol{\mathsf{e}}}) \setminus \{\mathbf{0}\}\} \notag\\
&\leq \overline{P}^{[\mathfrak{C}]}\left(\cancel{W}\middle|\mathsf{e}=e\right) \tag{5.17}\\
&= \frac{1}{q-1} \sum_{\mathbf{y}_{\boldsymbol{\mathsf{e}}} \in \mathbb{F}_q^e \setminus \{\mathbf{0}\}} \Pr\{\boldsymbol{\mathsf{H}}_{\boldsymbol{\mathsf{e}}} \mathbf{y}_{\boldsymbol{\mathsf{e}}} = \mathbf{0}\} \notag\\
&= \frac{1}{q-1} \sum_{w=1}^{e} \binom{e}{w} \Gamma_w \tag{5.18}
\end{align}
$$

In (5.18) the polynomial $\Gamma(\xi) = \sum_w \Gamma_w \xi^w$ is the average weight distribution of the considered parity-check ensemble and Γ_w denotes the number of codewords of weight w averaged over an ensemble. A detailed derivation of an expression for (5.17), which is based on the findings in [DPT$^+$02], can be found in [LPC13] for the special case of regular Gallager LDPC codes. The average weight distribution of a q-ary irregular parity-check ensemble with the design code rate ρ and the average variable node degree $\bar{\Lambda}$ is given by [KPD$^+$08]

$$\Gamma_w = \sum_{\kappa} \frac{\operatorname{coef}\Big(\big(A(\varsigma,\, \vartheta) B(\zeta)^{1-\rho}\big)^n,\, \vartheta^w \varsigma^\kappa \zeta^\kappa\Big)}{\binom{\bar{\Lambda} n}{\kappa} (q-1)^{\kappa - w}}, \tag{5.19}$$

with

$$A(\varsigma,\, \vartheta) = \prod_{i=2}^{d_{\mathrm{v,max}}} \left(1 + \vartheta \varsigma^i\right)^{\Lambda_i} \tag{5.20}$$

and

$$B(\zeta) = \prod_{j=2}^{d_{\mathrm{c,max}}} \left(\frac{(1 + (q-1)\zeta)^j + (q-1)(1-\zeta)^j}{q} \right)^{R_j}. \tag{5.21}$$

The above is a generalisation of the average weight distribution for binary parity-check ensembles from [Di04] to higher order Galois fields. The expression $\operatorname{coef}\big(B(\zeta),\, \zeta^i\big)$ denotes the coefficient of ζ^i in a polynomial $B(\zeta)$.

An upper bound on symbol level, given e symbol erasures in the encoded sequence, accordingly amounts to

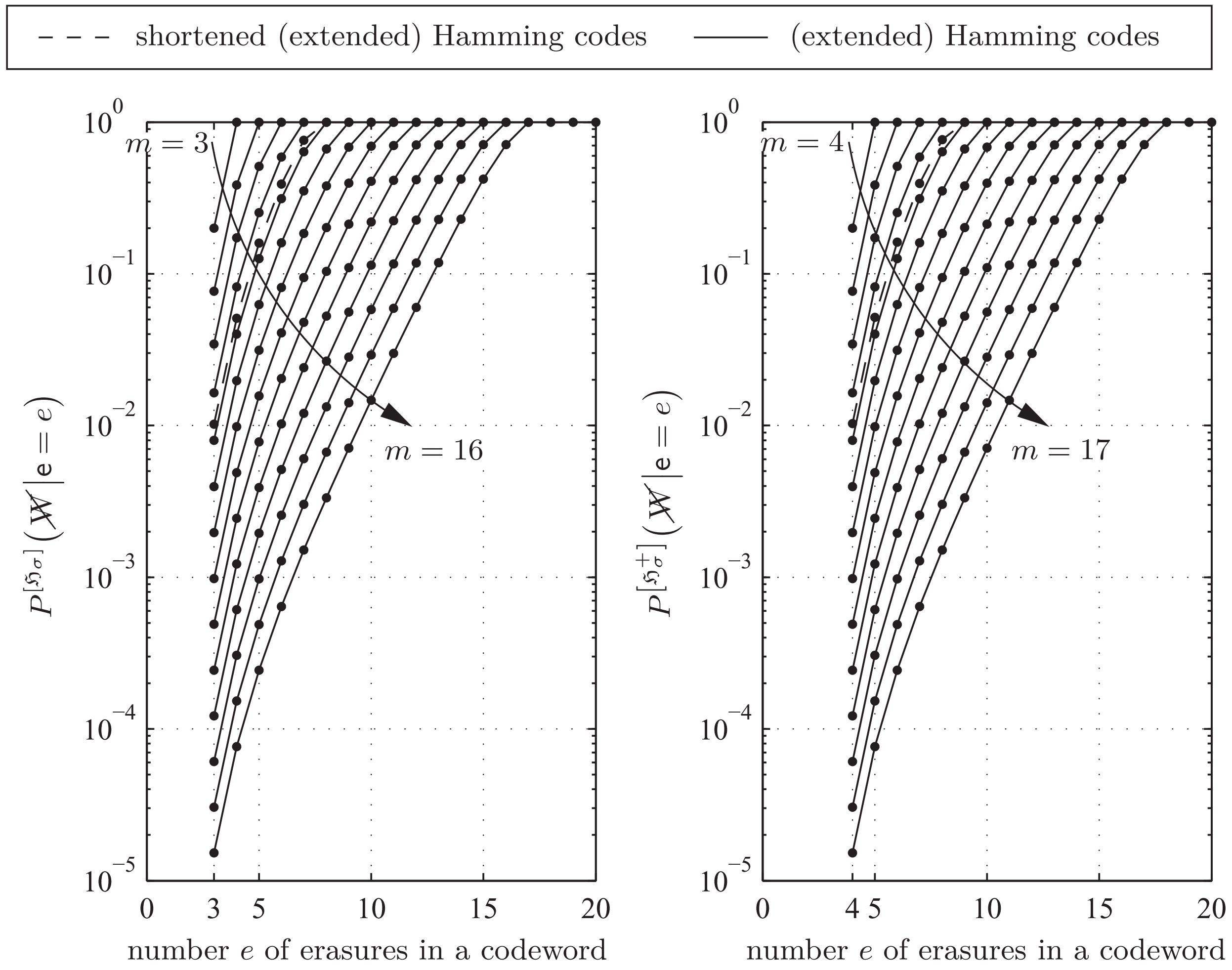
$$\overline{P}^{[\mathfrak{C}]}\big(\not{S} \,\big|\, \mathsf{e} = e\big) = \frac{1}{q-1} \sum_{w=1}^{e} \binom{e-1}{w-1} \Gamma_w. \tag{5.22}$$

For the special case of (sparse) random parity-check ensembles with density Δ an upper bound on word level is provided in [LPC13]

$$\overline{P}^{[\mathfrak{C}]}\big(\not{W} \,\big|\, \mathsf{e} = e\big) = \frac{1}{q-1} \sum_{w=1}^{e} \binom{e}{w} (q-1)^w \left(\frac{q-1}{q} \left(1 - \frac{q\Delta}{q-1}\right)^w + \frac{1}{q} \right)^m. \tag{5.23}$$

5.3 Numerical Evaluation and Examples

The decoding failure probabilities for (shortened) binary Hamming codes and for (shortened) extended Hamming codes are depicted in Figure 5.1 as a function of e, the number of erasures in a received encoded sequence. The two dashed lines depict



(a) Binary Hamming codes.

(b) Extended binary Hamming codes.

Figure 5.1: Decoding failure probabilities $P^{[\mathfrak{H}_\sigma]}\big(\cancel{\mathcal{W}}\,\big|\,\mathsf{e}=e\big)$ according to (5.8) and $P^{[\mathfrak{H}_\sigma^+]}\big(\cancel{\mathcal{W}}\,\big|\,\mathsf{e}=e\big)$ as in (5.14) of (shortened) (extended) binary Hamming codes with codeword lengths $n-\sigma$. The solid lines refer to unshortened codes with $m \in \{3, \dots, 16\}$, i.e. $n \in \{7, \dots, 65535\}$ and $n \in \{8, \dots, 65536\}$, respectively, while the dashed lines correspond to shortened codes with codeword length $n-\sigma = 100$, i.e. with $m = 7$ and $\sigma = 27$ as well as $m = 8$ and $\sigma = 28$, respectively.

the characteristics of the shortened codes which are tailored to match the here often used LT code input size of 100. Besides the well-known perfect recoverability of up to two or three erasures, respectively, it is interesting to see that even up to m erasures the recovery fails with a sufficiently small probability, particularly for e not too close to m. Here, MDS codes are not considered further, since their exact word erasure probability (5.1) is known and is merely a step function.

In Figure 5.2 upper bounds $\overline{P}^{[\mathfrak{C}]}\big(\cancel{\mathcal{W}}\,\big|\,\mathsf{e}=e\big)$ on the decoding failure probability of binary sparse random parity-check ensembles as given by (5.23) are depicted for different densities Δ and for $m = 8$ parity bits. This number of parity bits has

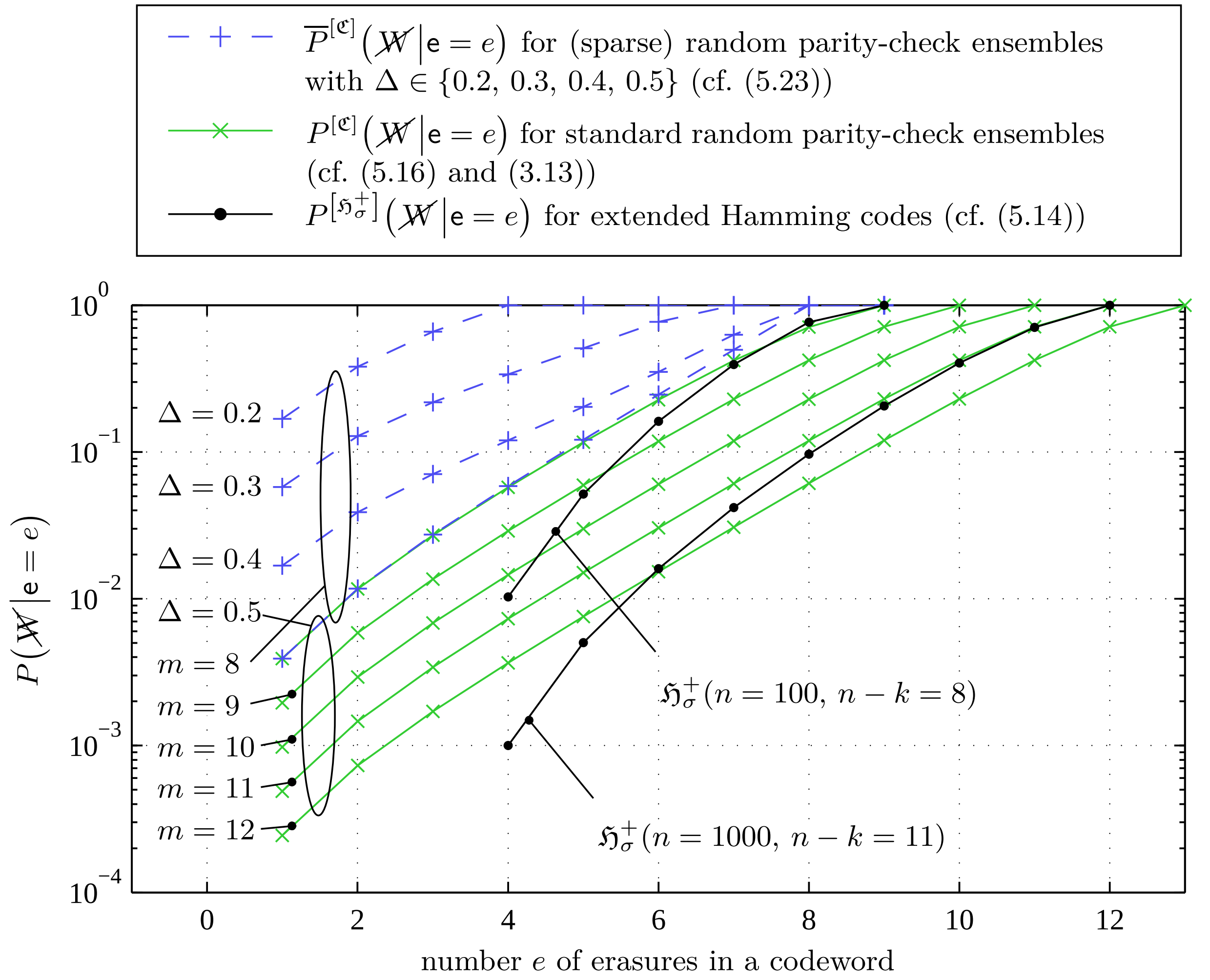


Figure 5.2: Decoding failure probabilities $P\big(\cancel{\mathcal{W}}\big|\mathsf{e}=e\big)$ for extended Hamming codes and for standard random parity-check ensembles, as well as upper bounds $\overline{P}^{[\mathfrak{C}]}\big(\cancel{\mathcal{W}}\big|\mathsf{e}=e\big)$ for (sparse) random parity-check ensembles with different densities. Note, the random ensembles' curves are independent of n and k.

been chosen to allow a fair comparison to the shortened extended Hamming code $\mathfrak{H}_\sigma^+$ with $n = 100$ and $k = 92$, which in the next chapter shall be employed as a precode. For the binary standard random ensemble ($\Delta = 0.5$) with 8 parity bits both the exact decoding failure probability as well as the upper bound is depicted, while for the ensembles with 9 to 11 parity bits only the exact decoding failure probability is shown, which can be determined by means of (5.16) and (3.13). The latter ensemble with $m = 11$ can be compared to the shortened extended Hamming code $\mathfrak{H}_\sigma^+$ with $n = 1000$ and $k = 989$ with the same number of parity bits.

Additionally, the upper bounds of two exemplary regular LDPC code ensembles are depicted in Figure 5.3 together with simulated erasure rates of six randomly chosen LDPC codes from the respective regular ensembles. For regular ensembles [Gal63, Mac97, Mac99], the number of parity bits is restricted to integer numbers given by $m = nd_c/d_v$, where d_c and d_v are the sole check node degree and variable node degree in a regular ensemble. Among the evaluated codes, the extended

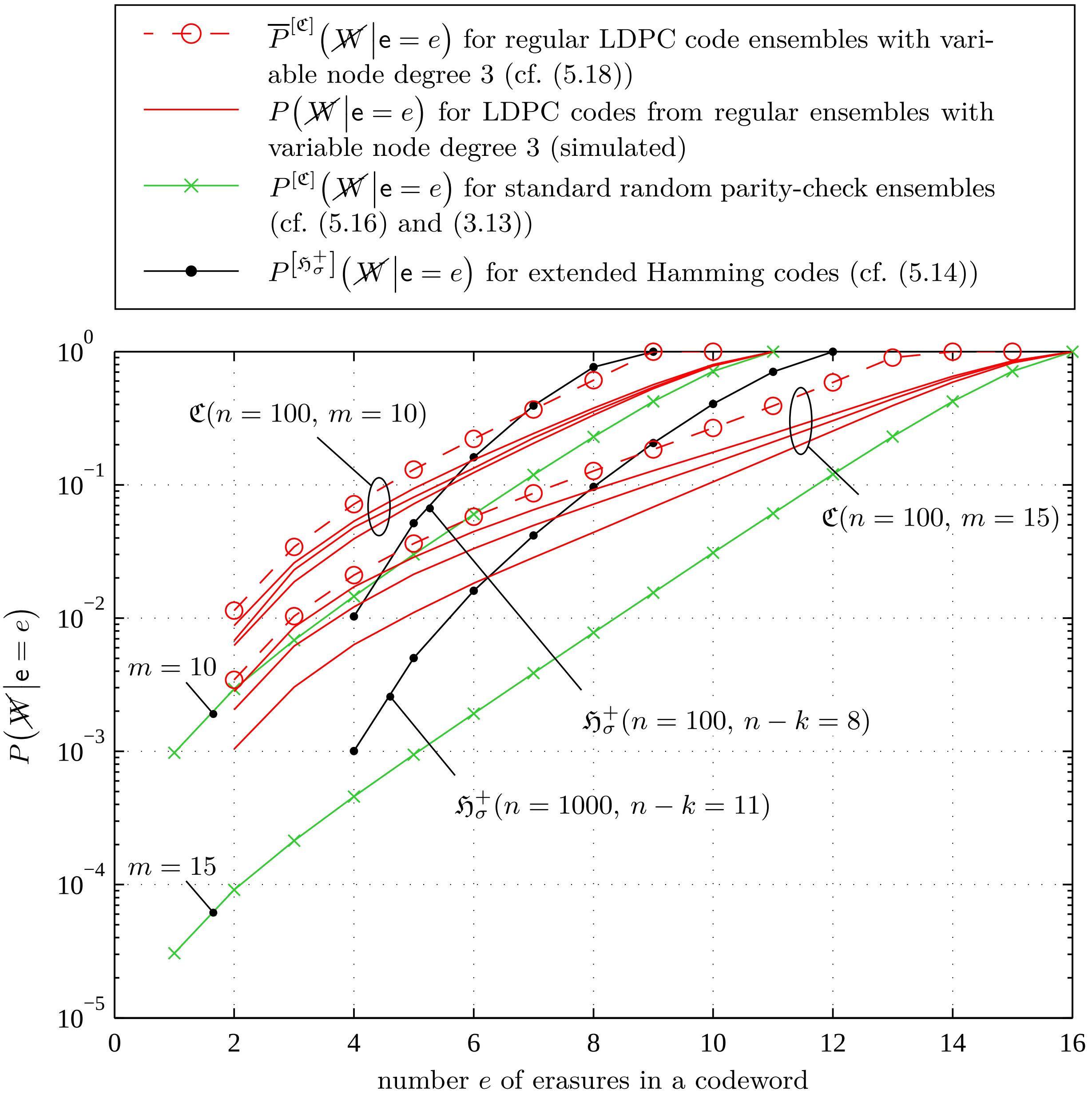


Figure 5.3: Upper bounds $\overline{P}^{[\mathfrak{C}]}\left(\cancel{W}\middle|\mathsf{e}=e\right)$ on the decoding failure probabilities of two regular LDPC code ensembles both with a variable node degree of 3 and a blocklength of $n=100$ bits. The number of parity-bits is $m=10$ and $m=15$, respectively. For three code realisations from each of the two ensembles the decoding failure rates have been simulated. Also depicted is the decoding failure probability $P^{[\mathfrak{C}]}\left(\cancel{W}\middle|\mathsf{e}=e\right)$ of two standard random ensembles with the same number of parity bits as the regular ensembles as well as $P^{[\mathfrak{H}_\sigma^+]}\left(\cancel{W}\middle|\mathsf{e}=e\right)$ for two extended Hamming codes. Note that the curves of the two random ensembles are independent of n and k.

Hamming codes perform better than the other considered parity-check codes with the same number of parity bits for almost all values of e. Therefore, but also due to their simplicity as well as for their available exact decoding failure probability they shall serve as example precodes in the next chapter.

Chapter 6

Raptor Code Ensembles

Raptor codes [Sho04, Sho06], or in simple terms precoded rateless codes, have been briefly introduced in Section 2.3. Note that in this chapter, the notation from Section 2.3 is used again (cf. also Figure 2.9), where dashed quantities relate to the information word $\mathbf{x}'$, i.e. the input to the precode. Quantities without a dash usually relate to the LT code.

The considered rateless component is typically an LT code, preferably of low density to ensure a low decoding complexity. As a low density is closely linked to a worryingly high erasure floor, at least one precode stage is usually required to lower the erasure floor to a suitable level. Employing a precode $\mathfrak{P}$ with code rate $\rho^{[\mathfrak{P}]} < 1$, however, induces a rate loss or conversely an additional overhead, i.e. the overall reception overhead is then $\gamma_{\mathrm{R}}^{[\mathfrak{P},\mathfrak{L}]} = \gamma_{\mathrm{R}}^{[\mathfrak{L}]}/\rho^{[\mathfrak{P}]}$. In order to minimise the rate loss, high-rate precodes are preferred.

Regarding the number of precodes, it highly depends on the application and as such on the size k' of the information word $\mathbf{x}'$ whether one or more precodes are required and which types. For large k' it is beneficial to use multiple high-rate precode stages, e.g. first an extended Hamming code and then an LDPC code as in [Sho06]. The decoding of large codes is usually performed iteratively for complexity reasons. However, as BP decoding of LDPC codes often suffers from so-called stopping sets[1], particularly from stopping sets of small size, an extended Hamming code with its minimal distance of four is employed to eliminate those stopping sets of size two and three, but also with a sufficiently high probability slightly larger ones [Sho06]. Note that if optimal (i.e. ML) decoding is feasible, for instance by using an efficient algorithm as described in Section 2.1.5, stopping sets are not an issue, since they appear only with suboptimal decoding.

For small k' and with optimal erasure decoding, as considered in this thesis, one precode is usually enough. For various reasons it also appears judicious to resort to simple yet efficient precodes like (extended) Hamming codes. First, to the best of the author's knowledge, there exist no short LDPC codes with a comparably high

[1] A stopping set is a subset of the variable nodes such that every check node connected to this set is connected to it at least twice.

code rate and equally good erasure correction properties. Second, due to their known erasure correction performance (cf. Chapter 5), particularly their known minimal Hamming distance $\delta_{\mathrm{H,min}}$, these codes can guarantee the correction of two or three erasures, respectively, within one information word $\mathbf{x}'$ and with a non-zero probability up to m erasures. So, since this chapter's goal is the exposition of methodologies for design and analysis of short Raptor codes ensembles, rather than the actual design of a record-breaking instance thereof, the simplicity and efficiency of such well-known example precodes like (extended) Hamming codes turns them into didactic prime examples. Nevertheless, if a higher minimal Hamming distance is required and a lower code rate is affordable, other codes can be good precode candidates, too, for instance some classes of (structured) LDPC codes, e.g. so-called finite geometry LDPC codes [KLF01]. And of course appropriately dimensioned MDS codes can also be considered as precodes.

6.1 Fundamentals

Before beginning with the essential part of this chapter, some basic terms and their usage shall be clarified first. The term dimension is often used in different ways [Str06]: for instance the vector $\mathbf{x} = (x_1, x_2, \ldots, x_k)^\mathsf{T}$ is said to be a k-dimensional vector in a k-dimensional vector space, e.g. $\mathbb{F}_q^k$. So this term is used both for vectors as well as for vector spaces or subspaces. In this thesis, the dimension operator $\dim(\cdot)$ will only be used for vector spaces or subspaces, i.e. $\dim(\mathbf{x})$ refers to the dimension of the subspace spanned by $\mathbf{x}$, which is a line and thus a one-dimensional subspace. Accordingly, two linearly independent vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ span a plane, i.e. a two-dimensional subspace and thus $\dim(\mathbf{x}_1, \mathbf{x}_2) = 2$.

Sometimes it is relevant to study mathematical objects not just by themselves but by taking into account the space that surrounds them: the ambient space. The statement "two lines, taken uniformly at random from the set of all possible lines, intersect almost surely" is true in a two-dimensional ambient space, but in a higher-dimensional ambient space the two lines are skew almost surely.

Aside from the span of a set of vectors, two further vector spaces and their dimensions are of interest: the parent vector space, here also denoted (global) ambient space, and a specific subspace, here denoted as minimal ambient space. The (global) ambient space $\mathcal{A}$, is the vector space with the maximum possible dimension which is k in the current example. It obviously contains all k-dimensional vectors and their spans. The minimal ambient space $\mathcal{A}^\star$ of the span of a set of vectors is the smallest subspace wherein this span can be embedded without transformation (e.g. rotation) w.r.t. the canonical basis of the global ambient space. An example is provided in Figure 6.1 to illustrate the different types of considered vector spaces.

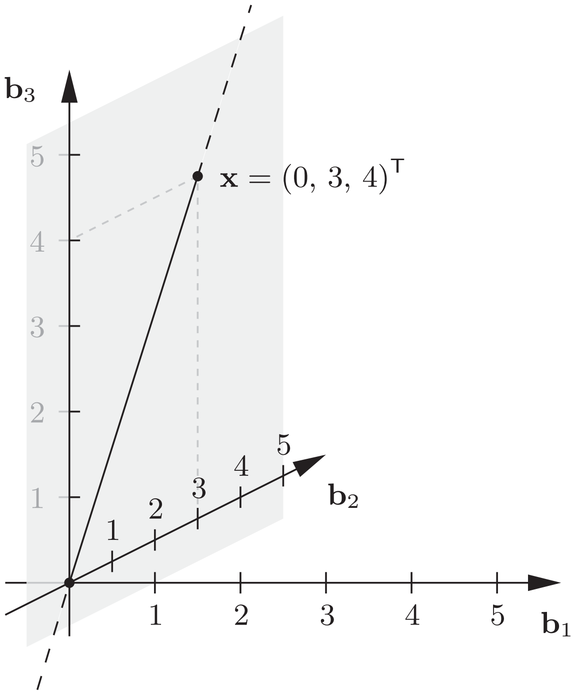


Figure 6.1: A three-dimensional vector $\mathbf{x} = (0,\, 3,\, 4)^\mathsf{T}$ in a three-dimensional (global) ambient space $\mathcal{A}(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ spanned by the basis $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_3$. The vector $\mathbf{x}$ spans a line that lies in the two-dimensional minimal ambient space $\mathcal{A}^\star(\mathbf{b}_2, \mathbf{b}_3)$ (grey plane) spanned by $\mathbf{b}_2$ and $\mathbf{b}_3$.

6.2 The Erasure Weight Profile

As already mentioned in Chapter 3, numerous publications deal with the (asymptotic) rank profile, i.e. the probability distribution of the rank, or equivalently with the nullity profile of random matrices over finite fields, i.e. the standard and the sparse random ensemble (disregarding expurgation). But apart from the relevant information whether a matrix has full column rank or not, in the context of erasure correction performance the knowledge of the exact rank or nullity does not give considerably more insight: while the nullity of a matrix $\mathbf{G}$ is equal to the number of linearly independent equations that are still required to make the system of random linear equations solvable, it is only a very unsharp lower bound on the number of unsolvable unknowns. Consider for instance solving the consistent system of linear equations $\mathbf{G}_\mathrm{R}\mathbf{x} = \mathbf{y}_\mathrm{R}$. The k unknowns $(x_1,\, x_2, \ldots,\, x_k)^\mathsf{T} = \mathbf{x} \in \mathbb{F}_q^k$ can be recovered if and only if $\mathbf{G}_\mathrm{R}$ has full column rank. If $\mathbf{G}_\mathrm{R}$ does not have full column rank, between nullity($\mathbf{G}_\mathrm{R}$) and k unknowns x_i remain unrecovered. This number, which is of particular interest when a precode is employed that can correct up to a certain number of erasures with a non-zero probability, is defined below together with some related quantities.

Definition 6.1. The number of unrecoverable unknowns in $\mathbf{x}$ after ML decoding a received codeword $\mathbf{y}_\mathrm{R}$, with $\mathbf{y}_\mathrm{R} = \mathbf{G}_\mathrm{R}\mathbf{x}$, is denoted erasure weight w_e in the following.

Corollary 6.2. The erasure weight w_e of an LT code is equal to the dimension of the minimum-dimensional subspace $\mathcal{A}^\star$ of $\mathbb{F}_q^k$ into which the kernel of $\mathbf{G}_\mathrm{R}$ can be embedded without transforming the kernel:

$$w_\mathrm{e} = \dim(\mathcal{A}^\star(\ker(\mathbf{G}_\mathrm{R}))), \tag{6.1}$$

where the mentioned ambient space containing $\ker(\mathbf{G}_\mathrm{R})$ is written as $\mathcal{A}^\star(\ker(\mathbf{G}_\mathrm{R}))$.

Definition 6.3. Accordingly, the erasure weight profile of an LT code ensemble $\mathfrak{L} = \left(\boldsymbol{\mathsf{G}}_\mathrm{R} \sim \Omega(\xi),\, \mathbb{F}_q^{n_\mathrm{R} \times k}\right)$ is defined as the probability distribution of w_e:

$$P^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{e} = w_\mathrm{e}) \triangleq \Pr\{\mathsf{w}_\mathrm{e} = w_\mathrm{e}\} = \Pr\{\dim(\mathcal{A}^\star(\ker(\boldsymbol{\mathsf{G}}_\mathrm{R}))) = w_\mathrm{e}\}, \tag{6.2}$$

for $0 \leq w_\mathrm{e} \leq k$.

Definition 6.4. Given the decoding failure probability distribution $P^{[\mathfrak{P}]}\left(\cancel{W}'\middle|\mathsf{w}_\mathrm{e} = w_\mathrm{e}\right)$ of a precode $\mathfrak{P}$ under ML decoding, conditioned on the number w_e of erasures in the intermediate codeword $\mathbf{x}$ and the erasure weight profile $P^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{e} = w_\mathrm{e})$ of an LT code ensemble $\mathfrak{L}$, the precoded (or Raptor code) erasure weight profile is determined by

$$P^{[\mathfrak{L},\mathfrak{P}]}\left(\cancel{W}', \mathsf{w}_\mathrm{e} = w_\mathrm{e}\right) \triangleq P^{[\mathfrak{P}]}\left(\cancel{W}'\middle|\mathsf{w}_\mathrm{e} = w_\mathrm{e}\right) \cdot P^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{e} = w_\mathrm{e}). \tag{6.3}$$

Corollary 6.5. Provided the precoded erasure weight profile $P^{[\mathfrak{L},\mathfrak{P}]}\left(\cancel{W}', \mathsf{w}_\mathrm{e}\right)$ of an LT code ensemble $\mathfrak{L}$ and a precode $\mathfrak{P}$, the ML decoding failure probability of the precoded LT code ensemble on word level equals to

$$P^{[\mathfrak{L},\mathfrak{P}]}\left(\cancel{W}'\right) = \sum_{w_\mathrm{e}=1}^{k} P^{[\mathfrak{L},\mathfrak{P}]}\left(\cancel{W}', \mathsf{w}_\mathrm{e} = w_\mathrm{e}\right). \tag{6.4}$$

Corollary 6.6. The corresponding ML decoding failure probability of a precoded LT code ensemble on symbol level is obtained by

$$P^{[\mathfrak{L},\mathfrak{P}]}\left(\cancel{S}'\right) = \sum_{w_\mathrm{e}=1}^{k} \frac{w_\mathrm{e}}{k} P^{[\mathfrak{L},\mathfrak{P}]}\left(\cancel{W}', \mathsf{w}_\mathrm{e} = w_\mathrm{e}\right). \tag{6.5}$$

In the special case that no precode is used, i.e. if $\mathbf{x} = \mathbf{I}_{k \times k}\mathbf{x}' = \mathbf{x}'$, (6.4) obviously results in

$$P^{[\mathfrak{L},\mathfrak{P}]}\left(\cancel{W}'\right) = P^{[\mathfrak{L}]}\left(\cancel{W}\right) = \sum_{w_\mathrm{e}=1}^{k} P^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{e} = w_\mathrm{e}) \tag{6.6}$$

and (6.5) in

$$P^{[\mathfrak{L},\mathfrak{P}]}\left(\cancel{S}'\right) = P^{[\mathfrak{L}]}\left(\cancel{S}\right) = \sum_{w_\mathrm{e}=1}^{k} \frac{w_\mathrm{e}}{k} P^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{e} = w_\mathrm{e}). \tag{6.7}$$

Subsequently, two examples are given to illustrate the meaning of the erasure weight and some related quantities.

Example 6.7. Let $\mathbf{G}_\mathrm{R} \in \mathbb{F}_2^{n_\mathrm{R} \times 7}$ be a realisation of the random matrix $\boldsymbol{\mathsf{G}}_\mathrm{R}$ and let the binary vector $\mathbf{x}_1 = (1, 1, 1, 0, 0, 0, 0)^\mathsf{T}$ be the only element in $\ker(\mathbf{G}_\mathrm{R}) \setminus \{\mathbf{0}\}$. Though the kernel has dimension one, i.e. $\mathrm{nullity}(\mathbf{G}_\mathrm{R}) = 1$, it occupies a three-dimensional subspace, i.e. $w_\mathrm{e} = \dim(\mathcal{A}^\star(\ker(\mathbf{G}_\mathrm{R}))) = 3$, or in other words, the support set of $\mathbf{x}_1$ has a cardinality of three. The three non-zero positions in $\mathbf{x}_1$ are those positions that cannot be recovered when solving $\mathbf{G}_\mathrm{R}\mathbf{x} = \mathbf{y}_\mathrm{R}$.

Example 6.8. Let $\mathbf{G}_\mathrm{R} \in \mathbb{F}_2^{n_\mathrm{R} \times 7}$ be a realisation of $\boldsymbol{\mathsf{G}}_\mathrm{R}$ and let the three binary vectors $\mathbf{x}_1 = (1, 1, 1, 0, 0, 0, 0)^\mathsf{T}$, $\mathbf{x}_2 = (1, 1, 0, 1, 0, 0, 0)^\mathsf{T}$ and $\mathbf{x}_3 = \mathbf{x}_1 + \mathbf{x}_2 = (0, 0, 1, 1, 0, 0, 0)^\mathsf{T}$ be the only vectors in $\ker(\mathbf{G}_\mathrm{R}) \setminus \{\mathbf{0}\}$ such that $\mathrm{nullity}(\mathbf{G}_\mathrm{R}) = 2$. Now, the number of unrecoverable unknowns is four, i.e. $w_\mathrm{e} = \dim(\mathcal{A}^\star(\ker(\mathbf{G}_\mathrm{R}))) = 4$, since there are four positions in $\mathbf{x}_1$ and $\mathbf{x}_2$ (not considering $\mathbf{x}_3$, as it is a linear combination of $\mathbf{x}_1$ and $\mathbf{x}_2$) that are non-zero in at least one of the two vectors. In other terms, w_e corresponds to the cardinality of the union of the kernel elements' support sets.

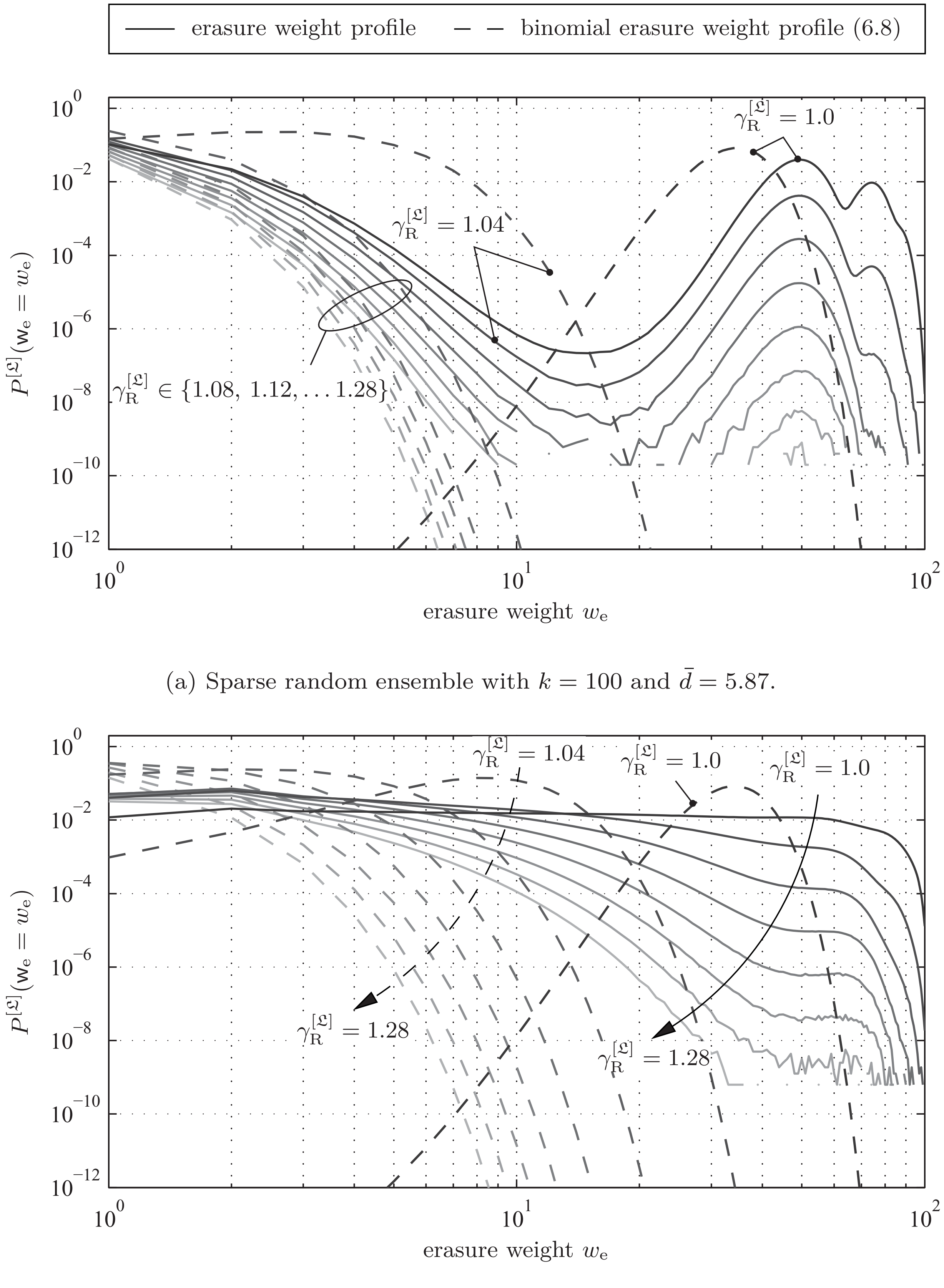
6.2.1 Binomial and Measured Erasure Weight Profiles

While the erasure weight w_e of a realisation $\mathbf{G}_\mathrm{R}$ of a random matrix $\boldsymbol{\mathsf{G}}_\mathrm{R}$ can be calculated just by solving the corresponding system of linear equations, the exact erasure weight profile of the ensemble $\boldsymbol{\mathsf{G}}_\mathrm{R}$ would have to be determined by solving all systems in the ensemble, which is by far too complex for reasonably sized ensembles. Since no method is known to determine $P^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{e} = w_\mathrm{e})$ analytically, it needs to be measured by solving a large number of realisations $\mathbf{G}_\mathrm{R}$ of $\boldsymbol{\mathsf{G}}_\mathrm{R}$, despite the rather high computational cost.

An attempt to model the erasure correction properties of a tandem ensemble consisting of an LT ensemble $\mathfrak{L}$ and some precode $\mathfrak{P}$ has already been undertaken in the literature. It has so far been assumed, e.g. in [RVF07, Lemmas 9 and 10] and [YLV$^+$13], that the symbol erasures after optimally decoding the LT code stage are independent, i.e. that the number w_e of residual symbol erasures after decoding the LT code stage follows a binomial distribution parametrised merely with the residual symbol erasure probability $P^{[\mathfrak{L}]}\big(\not\$\big)$:

$$P^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{e} = w_\mathrm{e}) = \binom{k}{w_\mathrm{e}} P^{[\mathfrak{L}]}\big(\not\$\big)^{w_\mathrm{e}} \big(1 - P^{[\mathfrak{L}]}\big(\not\$\big)\big)^{k - w_\mathrm{e}}. \tag{6.8}$$

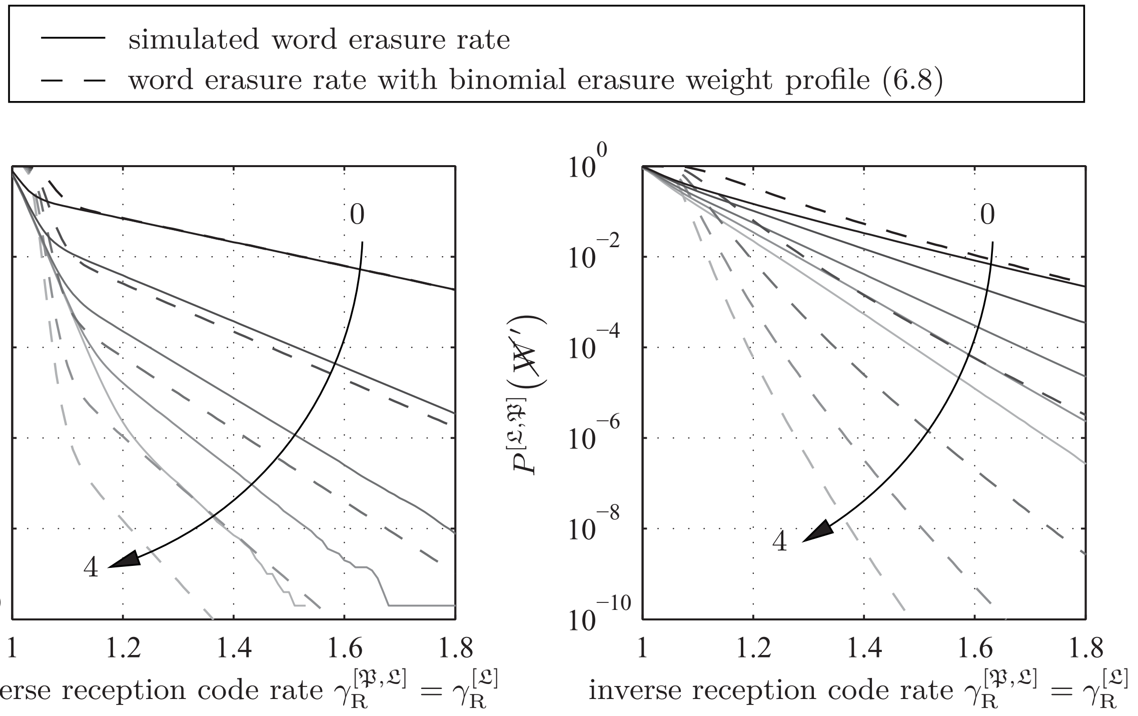
However, the assumption of independent residual symbol erasures after the LT code stage does not hold in general. It holds only if a sufficiently long interleaver is inserted between the precode and the LT code. Yet, a long interleaver induces an additional long delay and therefore it is not beneficial to use one in delay sensitive



(a) Sparse random ensemble with $k = 100$ and $\bar{d} = 5.87$.

(b) BP-optimised ensemble with $k = 100$, $\bar{d} = 5.87$ and $\Omega(\xi)$ from (2.8).

Figure 6.2: Erasure weight profiles (measured and binomial assumption) for two binary LT code ensembles for $\gamma_\mathrm{R}^{[\mathfrak{L}]} \in \{1.0, 1.04, 1.08, \ldots, 1.28\}$.



(a) Sparse random ensemble.

(b) BP-optimised ensemble, $\Omega(\xi)$ from (2.8).

Figure 6.3: Erasure correction performance of two genie-precoded binary LT code ensembles of intermediate codeword length $k = 100$ and with $\bar{d} = 5.87$. The genie precode corrects up to w_e symbol erasures within an ML decoded information word without rate loss, where $w_\mathrm{e} \in \{0, 1, \ldots, 4\}$. The dashed lines represent the respective erasure correction performance under the assumption of independent residual symbol erasures after decoding the LT stage [RVF07, YLV$^+$13], i.e. under the assumption of a binomial erasure weight profile as in (6.8).

applications. In terms of erasure correction performance it is more advantageous to increase the input size of the code than to employ an interleaver if a longer delay is affordable.

In Figure 6.2 the erasure weight profiles according to (6.8) are depicted as dashed lines for two LT code ensembles and for varying inverse reception code rates $\gamma_{\mathrm{R}}^{[\mathfrak{L}]}$. The corresponding measured erasure weight profiles are drawn as solid lines. Apparently, the true erasure weight profile depends strongly on the used row weight distribution, whereas the binomial erasure weight model only indirectly includes a weak dependency via the very high-level parameter $P^{[\mathfrak{L}]}\big(\not{\mathcal{S}}\big)$. It is very obvious that for neither code and for none of the values of $\gamma_{\mathrm{R}}^{[\mathfrak{L}]}$ does the binomial erasure weight model bear a very close resemblance with the measured erasure weight profile.

In Figure 6.3 the binomial erasure weight model has been used on two Raptor code ensembles consisting of the previous two LT code ensembles in combination with a genie precode. The idealised concept of a genie precode without rate loss is considered here, where the genie precode is assumed to recover up to $w_{\mathrm{e}} \in \{0, 1, \ldots, 4\}$ symbol erasures in the intermediate codeword at no cost. Also in this case it shows that the word erasure rates of the genie Raptor codes obtained from Monte Carlo simulations of the erasure weight profile have no likeness with the word erasure rates computed via (6.8). One can further observe that the stronger the genie precode the larger becomes the deviation of the modelled erasure rate from the simulated one. Evidently, the assumption of independent symbol erasures after LT decoding does not hold and as such (6.8) should not be used to predict the performance of Raptor code ensembles.

For the design of a Raptor code, it is not only important to obtain an LT code matrix with a high probability of achieving full rank, but given the possibility of employing a precode, it is crucial to find LT code ensembles with a suitable erasure weight profile. The erasure weight w_{e} corresponds to the number of unsolvable unknowns in the intermediate codeword $\mathbf{x}$ at the receiving end, i.e. the number of residual erasures in $\mathbf{x}$ after ML decoding of the LT code has been carried out. A precode can correct at best only $n'-k' = k-k'$ residual erasures in the intermediate codeword, which in case of a high-rate precode is a rather small number.

A good precodable LT code ensemble is thus characterised by low probabilities of high erasure weights, which are uncorrectable by a precode, whereas the probability of low erasure weights can be left to attain arbitrarily high values if such a degree of freedom allows to reduce the decoding complexity, for low erasure weights should be decodable with a sufficiently high probability by the chosen precode. In the two toy examples 6.7 and 6.8, a precode would have to be able to correct $w_{\mathrm{e}} = 3$ or $w_{\mathrm{e}} = 4$ erasures, respectively, within an intermediate codeword of length seven for an overall successful decoding.

Since measuring $P^{[\mathfrak{L}]}(\mathsf{w}_{\mathrm{e}} = w_{\mathrm{e}})$ by means of Monte Carlo simulations can be computationally expensive, another method will be proposed subsequently that allows first to approximate the erasure weight profile $P^{[\mathfrak{L}]}(\mathsf{w}_{\mathrm{e}} = w_{\mathrm{e}})$ of an LT code ensemble $\mathfrak{L}$. Then, given this approximation, it is possible to combine it with the erasure correction characteristic $P^{[\mathfrak{P}]}\left(\cancel{\mathscr{W}}'\middle|\mathsf{w}_{\mathrm{e}}\right)$ of an arbitrary high-rate precode $\mathfrak{P}$ and predict the erasure correction performance of the resulting Raptor code ensemble under ML decoding. The mentioned approximation is based on the upper bound on the word erasure probability in Section 3.3, and analogous to the upper bounds which can be used to model the erasure correction performance of an unprecoded LT code ensemble, the new approximation has the same quality for a precoded ensemble. If the upper bound is an accurate description of the plain LT code ensemble performance, then so is the new approximation for the precoded ensemble. This approximation can be remarkably accurate and can be computed with an extremely low complexity as will be demonstrated below.

6.3 The Kernel Weight Profile

In Examples 6.7 and 6.8 another quantity has been used besides the erasure weight w_{e}, namely the number of non-zero positions of the vectors in the respective kernel, i.e. their Hamming weights. This and some additional quantities, which shall serve to derive the previously mentioned approximation to the erasure weight profile, will be defined in the following:

Definition 6.9. The Hamming weight $\|\mathbf{x}\|_{\mathrm{H}}$ of a kernel element $\mathbf{x}$ shall be denoted kernel element weight $w_{\mathrm{k}} = \|\mathbf{x}\|_{\mathrm{H}}$.

Definition 6.10. The kernel of weight w_{k} of a matrix $\mathbf{G}_{\mathrm{R}}$ denotes the set of vectors $\mathbf{x}$ of Hamming weight w_{k} which are mapped to the null space

$$\ker(\mathbf{G}_{\mathrm{R}} \,|\, w_{\mathrm{k}}) = \{\mathbf{x} \,:\, \mathbf{G}_{\mathrm{R}}\mathbf{x} = \mathbf{0},\ \|\mathbf{x}\|_{\mathrm{H}} = w_{\mathrm{k}}\}. \tag{6.9}$$

Definition 6.11. The Hamming weights of all non-trivial kernel elements $\mathbf{x}_i$, $i \in \{1, \ldots, |\ker(\mathbf{G}_{\mathrm{R}})\setminus\{\mathbf{0}\}|\}$ shall be denoted kernel weight $\mathbf{w}_{\mathrm{k}} = (\|\mathbf{x}_1\|_{\mathrm{H}},\, \|\mathbf{x}_2\|_{\mathrm{H}}, \ldots)^{\mathsf{T}}$.

It is noteworthy that in Example 6.7 the erasure weight is equal to the kernel weight. The reason for this is that the kernel contains only one non-trivial element, i.e. $\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = 1$. Note that for higher order Galois fields, the $q-1$ non-trivial multiples of $\mathbf{x}$ are also in the kernel, but they have the same weight. So, in general, it is important that there exists only one linearly independent element in $\ker(\mathbf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}$, which is the case iff $\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = 1$.

In Example 6.8, however, while the erasure weight is four, the kernel weight is the triple $\mathbf{w}_{\mathrm{k}} = (\|\mathbf{x}_1\|_{\mathrm{H}},\, \|\mathbf{x}_2\|_{\mathrm{H}},\, \|\mathbf{x}_3\|_{\mathrm{H}})^{\mathsf{T}} = (3,\, 3,\, 2)^{\mathsf{T}}$, with $\ker(\mathbf{G}_{\mathrm{R}} \,|\, w_{\mathrm{k}} = 2) = \{\mathbf{x}_3\}$ and $\ker(\mathbf{G}_{\mathrm{R}} \,|\, w_{\mathrm{k}} = 3) = \{\mathbf{x}_1,\, \mathbf{x}_2\}$, since the kernel dimension is two (or more importantly greater than one), i.e. $\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = 2 > 1$, and it contains thus more than one linearly independent non-trivial element. Generally, the cardinality of the non-trivial kernel and the nullity are linked by $|\ker(\mathbf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}| = q^{\mathrm{nullity}(\mathbf{G}_{\mathrm{R}})} - 1$. So, in the second example, the scalar valued erasure weight obviously does not coincide with the kernel weight triple.

As it will be shown in Section 6.4, cases with $\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) > 1$ occur far less frequently than cases like Example 6.7 with $\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = 1$. So in most cases, there is only one linearly independent element in the kernel, and its weight w_{k} is then equal to the erasure weight w_{e}. This fact allows to approximate the erasure weight profile of an ensemble with another quantity which shall be denoted kernel weight profile. The latter, which has a close relation to the upper bound on the word erasure probability as given by (3.18), will be defined and derived in the following.

From the derivation of the upper bound on word level as given by (3.18) in Section 3.3, the expected cardinality of the non-trivial kernel of $\boldsymbol{\mathsf{G}}_\mathrm{R}$ is known as

$$\begin{aligned}
&\mathrm{E}\{|\ker(\boldsymbol{\mathsf{G}}_\mathrm{R}) \setminus \{\mathbf{0}\}|\} \\
&= \sum_{w=1}^{k} \binom{k}{w} (q-1)^{w} \left[\frac{1}{q} \sum_{d \in \mathcal{D}} \Omega_d \frac{\sum_{l=0}^{d} \binom{w}{l}\binom{k-w}{d-l}\left[1-(1-q)^{1-l}\right]}{\binom{k}{d}} \right]^{k\gamma_\mathrm{R}^{[\mathfrak{L}]}} . \qquad (6.10)
\end{aligned}$$

Using Definition 6.10, (6.10) can be reformulated to

$$\mathrm{E}\{|\ker(\boldsymbol{\mathsf{G}}_\mathrm{R}) \setminus \{\mathbf{0}\}|\} = \sum_{w_\mathrm{k}=1}^{k} \mathrm{E}\{|\ker(\boldsymbol{\mathsf{G}}_\mathrm{R} \,|\, \mathsf{w}_\mathrm{k} = w_\mathrm{k})|\}, \qquad (6.11)$$

where the summands constitute a scaled version of the kernel weight profile which is defined subsequently.

Definition 6.12. The kernel weight profile of an LT code ensemble $\mathfrak{L} = \left(\boldsymbol{\mathsf{G}}_\mathrm{R} \sim \Omega(\xi),\, \mathbb{F}_q^{n_\mathrm{R} \times k}\right)$ is the expected number of kernel elements $\mathbf{x}$ of Hamming weight w_k, i.e. the expected size of the kernel of weight w_k, excluding the kernel elements' $q-1$ non-trivial multiples

$$\begin{aligned}
&\frac{\mathrm{E}\{|\ker(\boldsymbol{\mathsf{G}}_\mathrm{R} \,|\, \mathsf{w}_\mathrm{k} = w_\mathrm{k})|\}}{q-1} \\
&\triangleq \binom{k}{w_\mathrm{k}} (q-1)^{w_\mathrm{k}-1} \left[\frac{1}{q} \sum_{d \in \mathcal{D}} \Omega_d \frac{\sum_{l=0}^{d} \binom{w_\mathrm{k}}{l}\binom{k-w_\mathrm{k}}{d-l}\left[1-(1-q)^{1-l}\right]}{\binom{k}{d}} \right]^{k\gamma_\mathrm{R}^{[\mathfrak{L}]}} , \qquad (6.12)
\end{aligned}$$

where $1 \leq w_\mathrm{k} \leq k$.

From Section 3.3 the upper bound on the residual word erasure probability after ML decoding is known and with the expressions from the previous section, the following relation holds between the sums over kernel and erasure weight profile:

$$\overline{P}^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right) = \sum_{w_\mathrm{k}=1}^{k} \frac{\mathrm{E}\{|\ker(\boldsymbol{\mathsf{G}}_\mathrm{R} \,|\, \mathsf{w}_\mathrm{k} = w_\mathrm{k})|\}}{q-1} \geq \sum_{w_\mathrm{e}=1}^{k} P^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{e} = w_\mathrm{e}) = P^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right). \qquad (6.13)$$

Unfortunately, this relation does not hold summand-wise, i.e. it does not hold for the kernel and the erasure weight profiles themselves. Nevertheless, like the upper bounds in Chapter 3 could be used as an approximation to the true erasure probabilities, the kernel weight profile can be used to approximate the erasure weight profile for $w_\mathrm{k} = w_\mathrm{e}$:

$$\widetilde{P}^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{k} = w_\mathrm{k}) \triangleq \widetilde{\Pr}\{\mathsf{w}_\mathrm{k} = w_\mathrm{k}\} = \frac{\mathrm{E}\{|\ker(\mathbf{G}_\mathrm{R} \,|\, \mathsf{w}_\mathrm{k} = w_\mathrm{k})|\}}{q-1} \approx P^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{e} = w_\mathrm{e}) \tag{6.14}$$

Note that $\widetilde{P}^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{k} = w_\mathrm{k})$ is not a true probability distribution, which is indicated by the tilde, and its sum may exceed one. In case that occurs, the sum, which corresponds to $\overline{P}^{[\mathfrak{L}]}\big(\not{W}\big)$, is clipped to one. Moreover, the kernel weight profile overestimates the probability of small erasure weights and underestimates the probability of large ones, i.e. for $w_\mathrm{k} = w_\mathrm{e}$

$$\begin{aligned} \widetilde{P}^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{k} = w_\mathrm{k}) &\gtrsim P^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{e} = w_\mathrm{e}) \quad \text{for small to medium } w_\mathrm{k} \text{ and } w_\mathrm{e} \\ \widetilde{P}^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{k} = w_\mathrm{k}) &\lesssim P^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{e} = w_\mathrm{e}) \quad \text{for large } w_\mathrm{k} \text{ and } w_\mathrm{e}. \end{aligned}$$

Now, consider for instance a binary matrix $\mathbf{G}_\mathrm{R}$ with $w_\mathrm{e} = 2$. The kernel weight can either be $\mathbf{w}_\mathrm{k} = (2)^\mathsf{T}$, too, or it can be $\mathbf{w}_\mathrm{k} = (1,\,1,\,2)^\mathsf{T}$. The first possibility occurs if there exist exactly two linearly dependent columns in $\mathbf{G}_\mathrm{R}$, while the second possibility arises from exactly two all-zero columns in $\mathbf{G}_\mathrm{R}$ if there are no further linearly dependent columns. In the previous case, $\mathrm{nullity}(\mathbf{G}_\mathrm{R}) = 1$, while in the latter $\mathrm{nullity}(\mathbf{G}_\mathrm{R}) = 2$. In general, $w_\mathrm{e} \geq \|\mathbf{w}_\mathrm{k}\|_\infty$, i.e. w_e is greater than or equal to the maximum entry in $\mathbf{w}_\mathrm{k}$.

So when approximating the erasure weight profile with the kernel weight profile, it should be kept in mind, that the kernel weight profile overestimates the probability of small erasure weights and underestimates the probability of large ones. The two one-entries in $\mathbf{w}_\mathrm{k} = (1,\,1,\,2)^\mathsf{T}$ for instance, have a small contribution to $\widetilde{P}^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{k} = 1)$, although the respective erasure weight is actually two. Nevertheless, this approximation is astonishingly good, since $\Pr\{\mathrm{nullity}(\mathbf{G}_\mathrm{R}) = 1\}$ is usually much larger than $\Pr\{\mathrm{nullity}(\mathbf{G}_\mathrm{R}) > 1\}$. And merely large nullities, which occur fairly seldom, contribute essentially to the misestimation.

Definition 6.13. Similarly to the precoded erasure weight profile, the precoded kernel weight profile is given by

$$\widetilde{P}^{[\mathfrak{L},\mathfrak{P}]}\big(\not{W}', \mathsf{w}_\mathrm{k} = w_\mathrm{k}\big) \approx P^{[\mathfrak{P}]}\big(\not{W}' \big| \mathsf{w}_\mathrm{e} = w_\mathrm{e}\big) \cdot \widetilde{P}^{[\mathfrak{L}]}(\mathsf{w}_\mathrm{k} = w_\mathrm{k}), \tag{6.15}$$

for $w_\mathrm{e} = w_\mathrm{k}$ and given the decoding failure probability distribution $P^{[\mathfrak{P}]}\big(\not{W}' \big| w_\mathrm{e}\big)$ of a precode $\mathfrak{P}$ under ML decoding, conditioned on the number w_e of erasures in an intermediate codeword $\mathbf{x}$.

It is assumed, that the precode can eliminate an intermediate codeword of weight w_k from the kernel if it is able to correct w_e erasures in the intermediate codeword.

Again it shall be emphasised that this assumption holds exactly only in the case of $\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = 1$. And also in the case of the precoded profiles, the precoded kernel weight profile will be used as an approximation for the precoded erasure weight profile for $w_{\mathrm{e}} = w_{\mathrm{k}}$

$$P^{[\mathfrak{L},\mathfrak{P}]}\big(\cancel{\mathcal{W}}', \mathsf{w}_{\mathrm{e}} = w_{\mathrm{e}}\big) \approx \widetilde{P}^{[\mathfrak{L},\mathfrak{P}]}\big(\cancel{\mathcal{W}}', \mathsf{w}_{\mathrm{k}} = w_{\mathrm{k}}\big). \tag{6.16}$$

6.4 The Nullity Profile

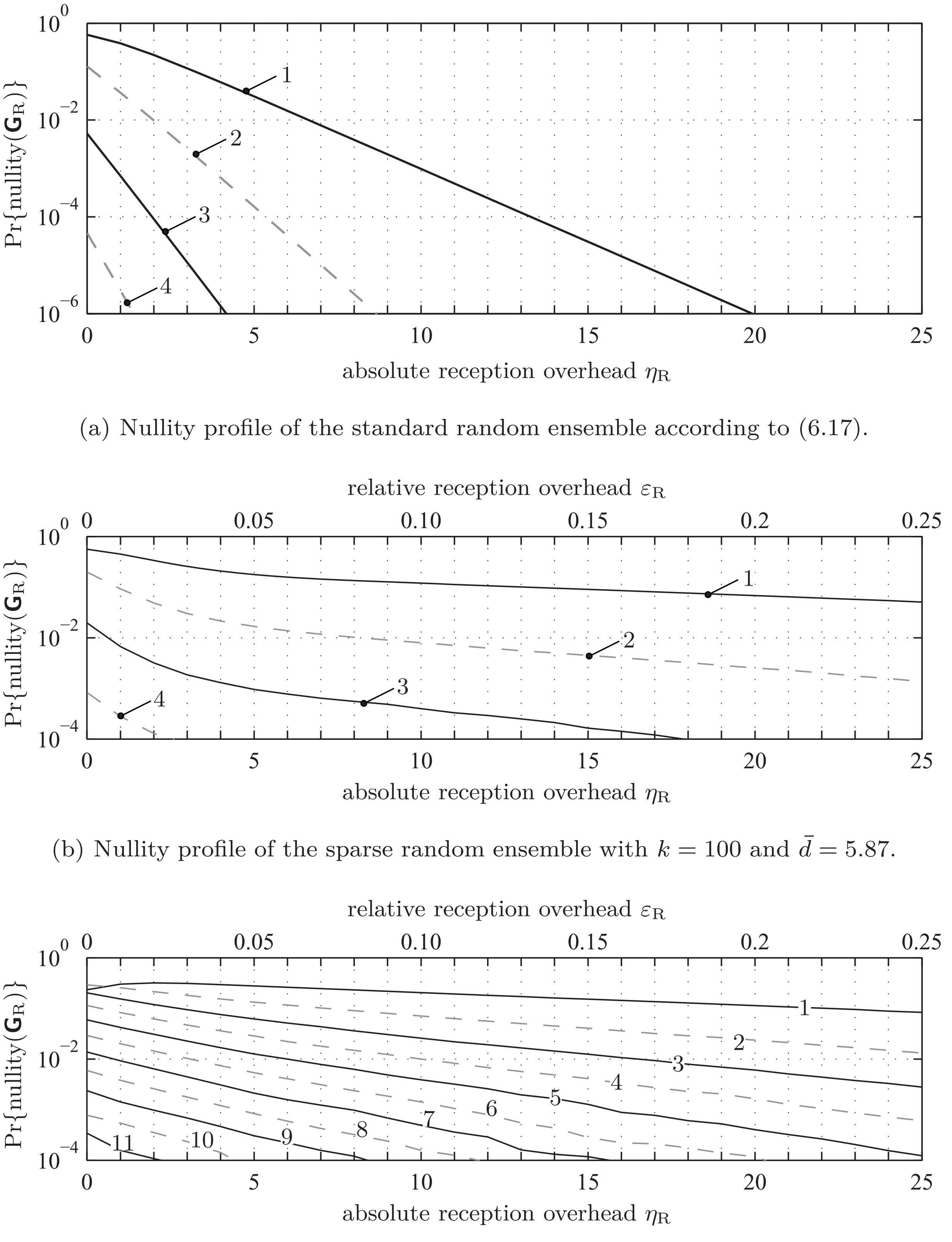
The nullity profile or equivalently the rank profile is only known for the standard random ensemble. It is obtained as the quotient of (3.10) and (3.12)

$$\begin{aligned}
\Pr\{\mathrm{rank}(\mathbf{G}_{\mathrm{R}}) = r\} &= \Pr\{\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = o\} \quad \text{with } o = k - r \\
&= \frac{N(n_{\mathrm{R}} \times k,\, r)}{\sum_{r=0}^{k} N(n_{\mathrm{R}} \times k,\, r)} \\
&= \frac{\begin{bmatrix} n_{\mathrm{R}} \\ r \end{bmatrix}_q \big(q^k - 1\big)\big(q^k - q\big)\big(q^k - q^2\big)\ldots\big(q^k - q^{r-1}\big)}{q^{k n_{\mathrm{R}}}}.
\end{aligned} \tag{6.17}$$

Remark 6.14. To the best of the author's knowledge the complete rank profile of random matrices is given analytically only for the standard random ensemble. For all other ensembles the determination of the complete rank profile is still an open problem. Even the exact probability of full rank could only be derived for the expurgated random ensemble, as shown in Section 3.2.2.

The rank and nullity profiles of the standard random ensemble are essentially independent of the input size k when evaluated for not too tiny values of k. The nullity profile of the standard random ensemble is depicted in Figure 6.4(a) as a function of the absolute reception overhead η_{R}. For two other binary LT ensembles of input size $k = 100$, namely the sparse random ensemble and the BP-optimised ensemble both with an average row weight $\bar{d} = 5.87$, which have yet repeatedly served as examples, the nullity profile has been measured and depicted in Figures 6.4(b) and 6.4(c). These two ensembles have been assigned a fixed input size, since their performance is not independent of k. Therefore, the relative reception overhead ε_{R} is provided, too. In contrast to the exponential decrease of $\Pr\{\mathrm{nullity}(\mathbf{G}_{\mathrm{R}})\}$ of the standard ensemble as a function of the overhead, the terms of the nullity profile of the sparse random ensemble seem to exhibit a waterfall and a floor region. Such a structure cannot be clearly identified in case of the BP-optimised ensemble.

Moreover, while $\Pr\{\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = o_1\} \gg \Pr\{\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = o_2\}$ if $o_1 < o_2$ for all overheads for the random ensembles, this is not the case for the BP-optimised ensemble, where for small overhead values $\Pr\{\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = 1\} < \Pr\{\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = 2\}$. And also $\Pr\{\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = o\}$ does not decrease as fast with



(a) Nullity profile of the standard random ensemble according to (6.17).

(b) Nullity profile of the sparse random ensemble with $k = 100$ and $\bar{d} = 5.87$.

(c) Nullity profile of the BP-optimised ensemble (2.8) with $k = 100$ and $\bar{d} = 5.87$.

Figure 6.4: Nullity profiles of three binary ensembles. Subplots (b) and (c) are obtained by means of Monte Carlo simulations. The numbers in the figures denote the respective nullities.

o as in the cases of the random ensembles. But note that when the density of the sparse ensemble is decreased further such that it degenerates, it can also occur that $\Pr\{\text{nullity}(\mathbf{G}_\mathrm{R}) = o_1\} < \Pr\{\text{nullity}(\mathbf{G}_\mathrm{R}) = o_2\}$ for small overhead values if $o_1 < o_2$.

Nevertheless, it has been exemplarily demonstrated that for codes of interest $\Pr\{\text{nullity}(\mathbf{G}_\mathrm{R}) = o_1\} \gg \Pr\{\text{nullity}(\mathbf{G}_\mathrm{R}) = o_2\}$ if $o_1 < o_2$, which is required as a justification for approximating the (precoded) erasure weight profile with the (precoded) kernel weight profile with sufficient accuracy.

6.5 Numerical Evaluation and Monte Carlo Simulations

For the two example LT code ensembles used throughout this chapter, cf. Figures 6.2 and 6.3, the measured erasure weight profiles have already been depicted in Figure 6.2 where it has been shown that the erasure weight profile does not follow a binomial distribution as repeatedly assumed in the literature. In Figures 6.5(a) and 6.6(a) the measured profiles have been redrawn together with the corresponding kernel weight profiles, which acts as an upper bound on the erasure weight profile for not too high weights. The deviation of the kernel weight profile from the erasure weight profile or more precisely the reason for their congruence has already been explained in Section 6.3. In Figure 6.5(a), i.e. for the sparse random LT code ensemble, the two profiles are almost congruent for a very large range of weights, particularly for low weights, while the profiles of the BP-optimised LT code ensemble in Figure 6.6(a) differ significantly for small inverse reception code rates $\gamma_\mathrm{R}^{[\mathfrak{L}]}$, but they quickly approach each other as $\gamma_\mathrm{R}^{[\mathfrak{L}]}$ increases.

This behaviour is in line with that of the upper bounds on word or symbol level as a function of $\gamma_\mathrm{R}^{[\mathfrak{L}]}$. The upper bounds for the sparse random ensemble for instance are extremely good approximations of the residual erasure rates for all $\gamma_\mathrm{R}^{[\mathfrak{L}]}$, while in the case of the BP-optimised ensemble the upper bound is less suited to approximate the residual erasure rates at low $\gamma_\mathrm{R}^{[\mathfrak{L}]}$ close to one, but the approximation accuracy improves with increasing $\gamma_\mathrm{R}^{[\mathfrak{L}]}$. As a rule of thumb it can be said that the upper bounds are very good approximations of the residual erasure probabilities. Consequently, the approximation of the erasure weight profile by the kernel weight profile is feasible if the upper bounds exhibit a certain shape, i.e. the waterfall region should resemble that of the standard random ensemble and when reaching the proximity of the lower bound, the respective upper and lower bounds should quickly converge.

While the behaviour of the upper bound serves merely as an indicator of whether to expect a good or a bad approximation quality by the kernel weight profile, the actual reasoning behind the approximation attempt is the fact that erasure and kernel weight are the same if $\text{nullity}(\mathbf{G}_\mathrm{R}) = 1$. In Section 6.4 the nullity profile of LT code ensembles has been introduced and its characteristics have been illustrated by means of three realistic examples. The exhibited predominance of

$\Pr\{\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = 1\}$, where it applies, results in a high accuracy of the introduced approximation of the erasure weight profile by means of the kernel weight profile, as for instance for the sparse random LT code ensemble. Where $\Pr\{\mathrm{nullity}(\mathbf{G}_{\mathrm{R}}) = 1\}$ is less predominant, as it is the case with the BP-optimised ensemble, the approximation accuracy certainly suffers.

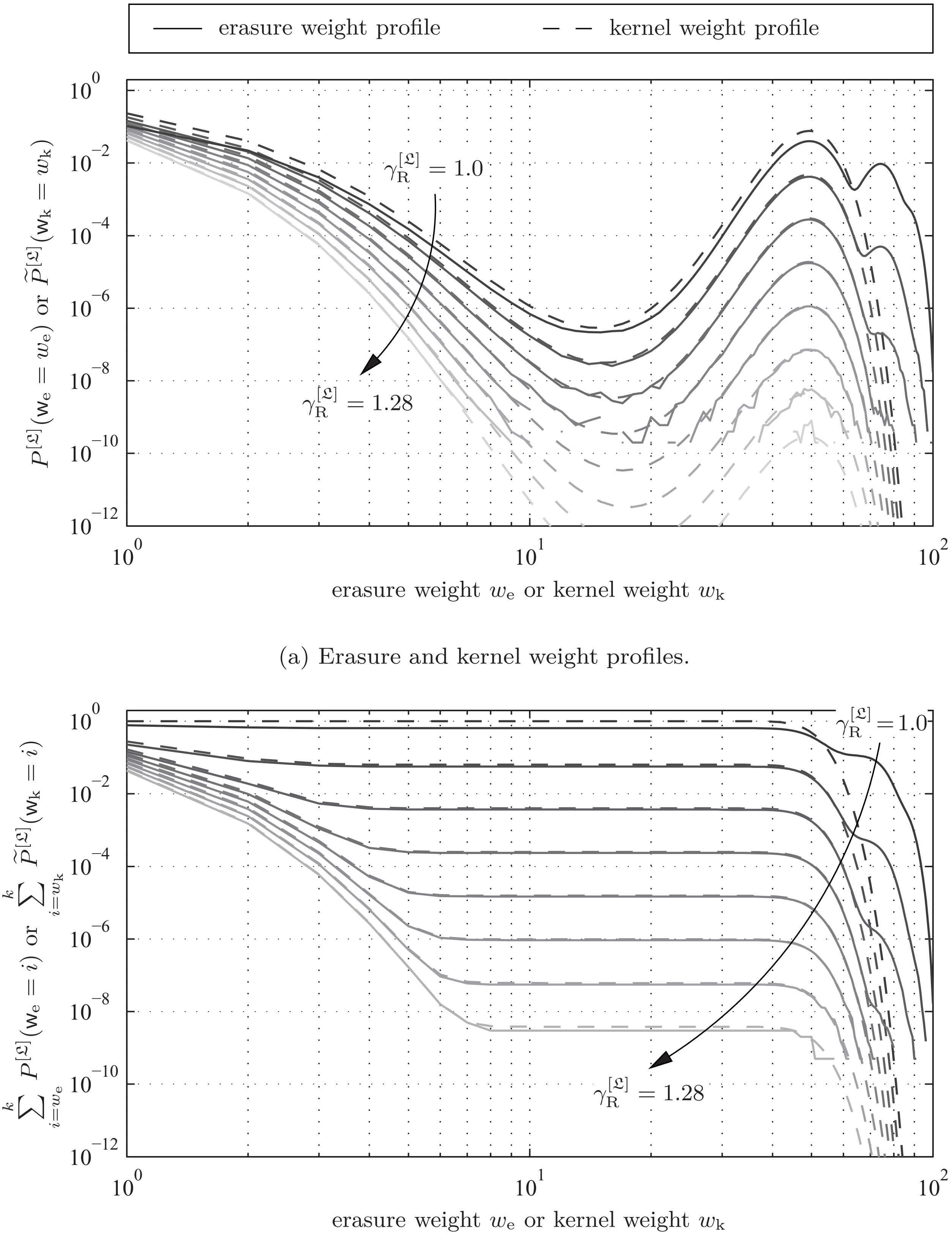
In Figures 6.5(b) and 6.6(b) the respective complementary cumulated erasure and kernel weight profiles are shown. A steep descent at low weights indicates a good suitability for high-rate precoding, by which a few residual erasures are correctable with high probability. As the complementary cumulated profile flattens out, a precode that is able to correct erasures in that range of weights has no advantage over a precode that can correct only up to the beginning of the flat region. The contrary is even the case if the stronger precode has a smaller rate, since then the rate loss is larger.

Assuming a genie precode that can correct all residual erasures up to a weight of w_{e}, the complementary cumulated erasure weight profile at $w_{\mathrm{e}} + 1$ yields the residual word erasure probability after precoding. Alike, the complementary cumulated kernel weight profile results in a pseudo upper bound thereof if the number of correctable erasures $w_{\mathrm{e}} \geq 1$ is not too large. If no precode is applied, i.e. if the number of correctable erasures w_{e} is zero, the complementary cumulated kernel weight at $w_{\mathrm{e}} = 1$ is a true upper bound.

For the two example codes the resulting erasure rates as well as the (pseudo) upper bounds are depicted in Figure 6.7 using a genie precode that can correct up to $w_{\mathrm{e}} \in \{0, 1, \ldots, 4\}$ erasures. As expected from the strong congruence of the kernel and erasure weight profiles, the erasure correction performance of the sparse random ensemble is very accurately predicted by the (pseudo) upper bounds. The performance of the BP-optimised ensemble in contrast thereto is only accurately modelled by the (pseudo) upper bounds for larger values of $\gamma_{\mathrm{R}}^{[\mathfrak{L}]}$. From both genie-precoded ensembles it becomes clear that the pseudo upper bounds are only as good a description of the real erasure rate as the upper bounds of the unprecoded ensembles.

In Figures 6.8(a) and 6.8(b) the residual erasure probabilities of the two example LT code ensembles with an intermediate codeword length of $k = 100$ are depicted together with those of the precoded ensembles using an appropriately shortened extended Hamming code. For the unprecoded ensembles upper and lower bounds are also given, while for the precoded ensembles only the pseudo upper bounds are drawn, since proper (pseudo) lower bounds have not yet been found.

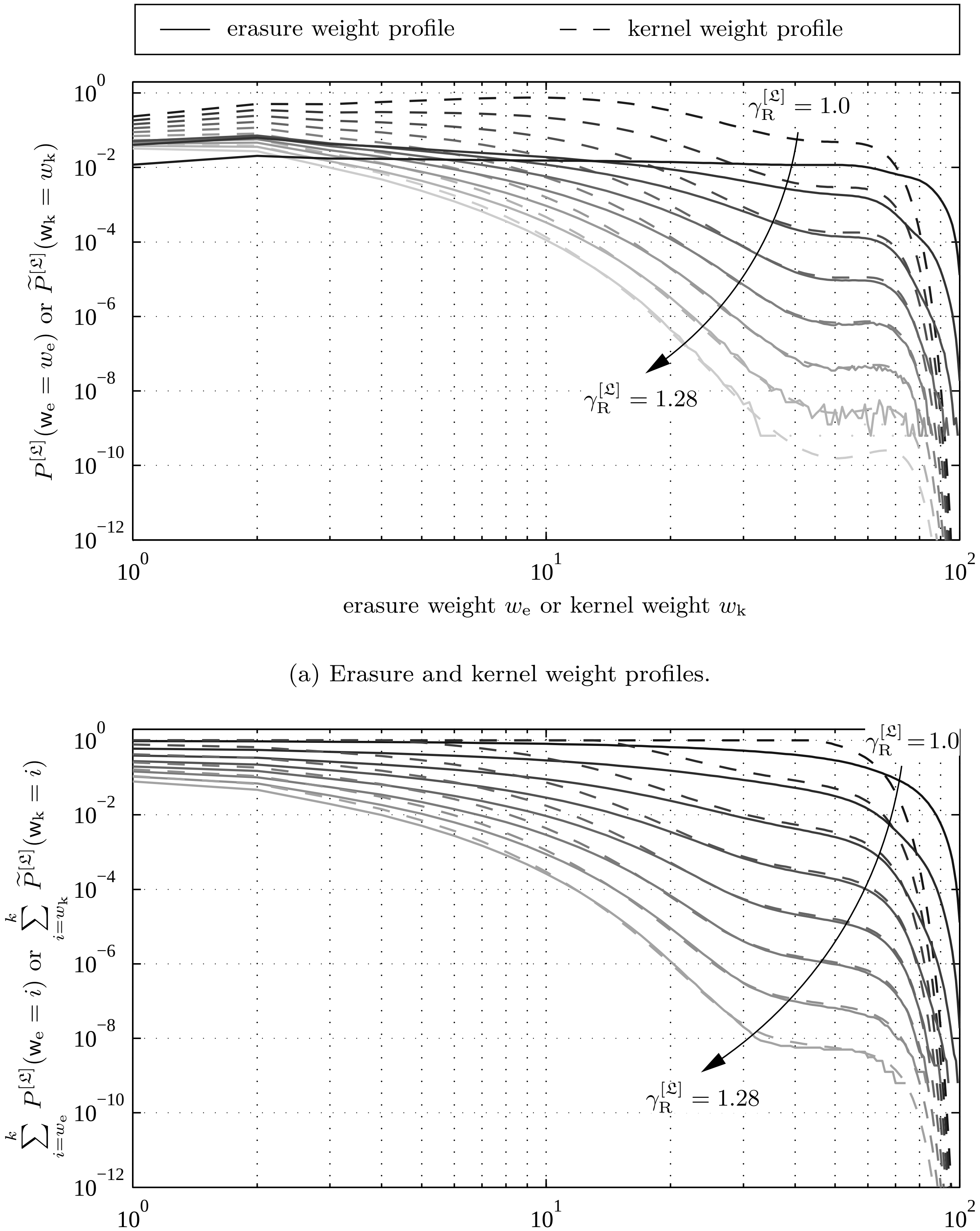
Note that for a real precode, whose code rate is lower than one, the curves are shifted to the right. The waterfall region is dominated by high-weight erasures which a high-rate precode cannot remove. With increasing $\gamma_{\mathrm{R}}^{[\mathfrak{L}]}$ the amount of high-weight erasures decreases much faster than the amount of low-weight erasures, which can be observed in the respective weight profiles. At the same time



(a) Erasure and kernel weight profiles.

(b) Complementary cumulated erasure and kernel weight profiles.

Figure 6.5: Erasure and kernel weight profiles and the respective complementary cumulants (i.e. from right to left) for $\gamma_\mathrm{R}^{[\mathfrak{L}]} \in \{1.0, 1.04, 1.08, \ldots, 1.28\}$ for the binary sparse random LT code ensemble with $k = 100$ and $\bar{d} = 5.87$.



(a) Erasure and kernel weight profiles.

(b) Complementary cumulated erasure and kernel weight profiles.

Figure 6.6: Erasure and kernel weight profiles and the respective complementary cumulants (i.e. from right to left) for $\gamma_{\mathrm{R}}^{[\mathfrak{L}]} \in \{1.0, 1.04, 1.08, \ldots, 1.28\}$ for the BP-optimised LT code ensemble with $k = 100$, $\bar{d} = 5.87$ and $\Omega(\xi)$ from (2.8).

the precode gets more and more useful, since it obtains an increasing fraction of correctable low-weight erasures from the LT decoder. It only fails at the fewer remaining high-weight erasures and thus prolongs the waterfall region such that the erasure floor reaches a much lower level, where the latter is also much steeper than the unprecoded one.

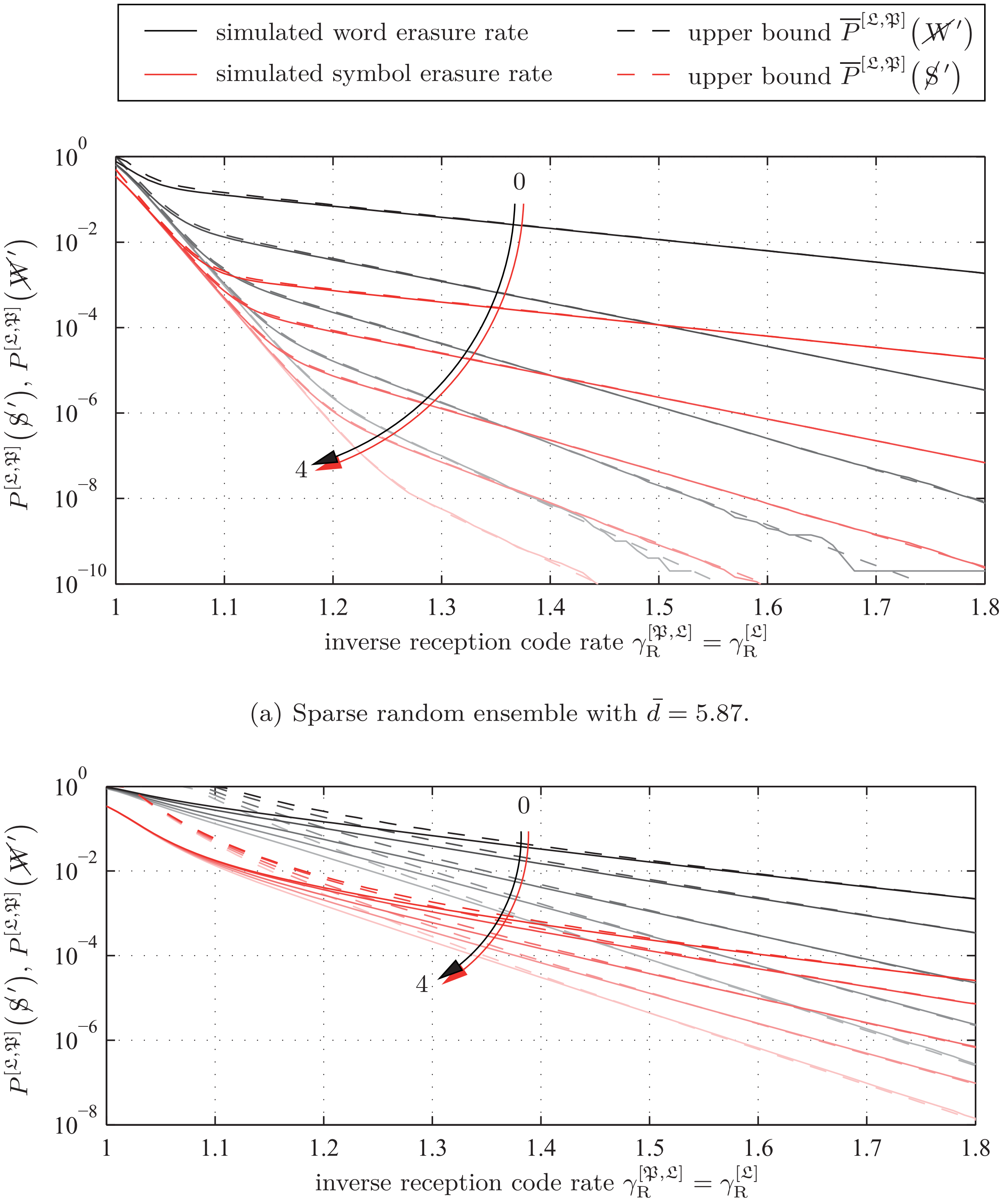
As above with the genie-precoded ensembles, the pseudo upper bounds match the measured erasure rates very well. The effect of this precode on the erasure and kernel weight profiles can be observed in Figure 6.9. The profiles are cut off at a weight of $\delta_{\mathrm{H,min}} = 3$ and lowered up to a weight of $m = 8$. Accordingly, the complementary cumulated profiles saturate at $\delta_{\mathrm{H,min}} + 1$.

As a final example, the erasure correction performance of two larger (precoded) LT code ensembles is provided in Figures 6.8(c) and 6.8(d). The row weight has been kept constant, while the intermediate codeword length has been increased to $k = 1000$ and an appropriately sized shortened extended Hamming code is used as an exemplary precode. The unprecoded ensembles, again a sparse random ensemble and the BP-optimised ensemble, are slightly degenerated, i.e. $\bar{d} < \ln k$, and thus the waterfall regions of the upper bounds stop significantly above the lower bounds and approach them at a slower pace. This effect can be particularly well observed for the sparse random ensemble. Yet, the pseudo upper bounds are again of the same quality as the upper bounds of the unprecoded ensembles and allow a sufficiently accurate performance assessment of the precoded ensembles, especially at higher $\gamma_{\mathrm{R}}^{[\mathfrak{L}]}$ or $\gamma_{\mathrm{R}}^{[\mathfrak{P},\mathfrak{L}]}$. Since the used precode has a much higher code rate than the one in the short ensemble, the rate loss is significantly lower.

6.6 Conclusions

Three quantities, i.e. the erasure weight profile, the kernel weight profile and the nullity profile have been introduced in this chapter to characterise the erasure correction performance of LT code ensembles on a finer scale than merely the residual erasure probability on word or on symbol level or the respective bounds thereon. As the erasure weight profile, the most valuable quantity of the three, cannot be determined analytically, although incorrect attempts hereto exist in the literature, it has to be obtained via extensive Monte Carlo simulations.

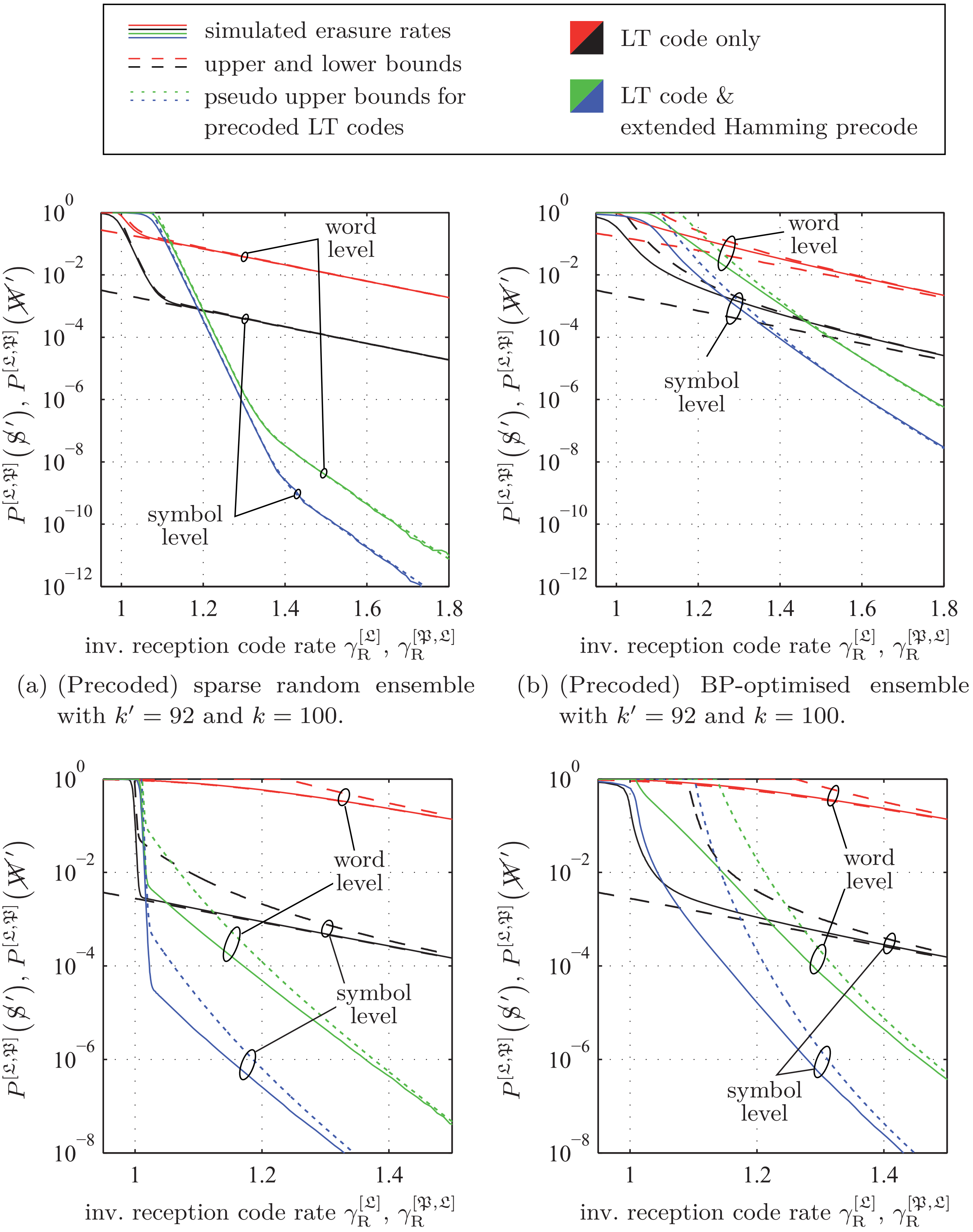
Given that the nullity profile of the LT code ensemble fulfils the condition that the probability of large nullities is small, which should be the case for good LT code ensembles, this allows to use the kernel weight profile, which can be obtained from the expression of the previously determined upper bound on word level, as an approximation to the erasure weight profile and thus (almost) redundantise its simulative determination. Together with the conditional decoding failure probability of the respective precode this finally allows to quickly and accurately assess the compound erasure correction performance of Raptor code ensembles under optimal decoding.



(a) Sparse random ensemble with $\bar{d} = 5.87$.

(b) BP-optimised ensemble with $\Omega(\xi)$ from (2.8) and $\bar{d} = 5.87$.

Figure 6.7: Erasure correction performance of two genie-precoded binary LT code ensembles of input size $k = 100$ and with $\bar{d} = 5.87$. The genie precode corrects up to w_{e} symbol erasures within an ML decoded information word without rate loss, where $w_{\mathrm{e}} \in \{0, 1, \ldots, 4\}$. The upper bounds for $w_{\mathrm{e}} \in \{1, \ldots, 4\}$ are in fact pseudo upper bounds derived from the respective kernel weight profiles depicted in Figures 6.5(a) and 6.6(a).



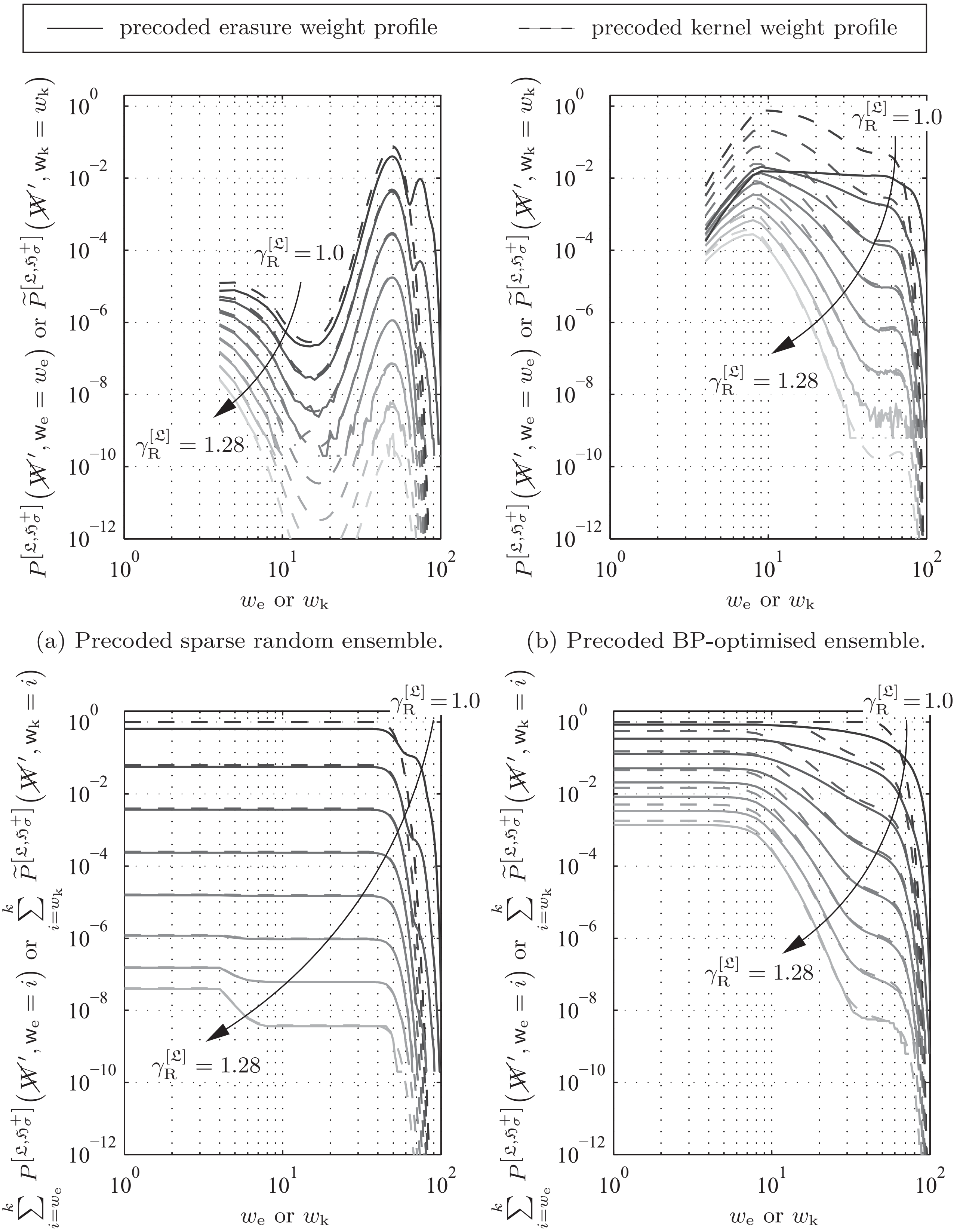
(a) (Precoded) sparse random ensemble with $k' = 92$ and $k = 100$.

(b) (Precoded) BP-optimised ensemble with $k' = 92$ and $k = 100$.

(c) (Precoded) sparse random ensemble with $k' = 989$ and $k = 1000$.

(d) (Precoded) BP-optimised ensemble with $k' = 989$ and $k = 1000$.

Figure 6.8: Bounds and simulated erasure rates of (precoded) binary LT codes with (intermediate) input sizes $k = 100$ and $k = 1000$, all with $\bar{d} = 5.87$. The precodes are appropriately shortened extended Hamming codes with $k' = 92$ and $k' = 989$, respectively. The rate loss of the precoded ensembles appears as a right-shift of the respective curves.



(a) Precoded sparse random ensemble.

(b) Precoded BP-optimised ensemble.

(c) Precoded sparse random ensemble.

(d) Precoded BP-optimised ensemble.

Figure 6.9: (Complementary cumulated) precoded erasure and kernel weight profiles for $\gamma_{\mathrm{R}}^{[\mathfrak{L}]} \in \{1.0, 1.04, 1.08, \ldots, 1.28\}$ for a sparse random LT ensemble (left) and a BP-optimised ensemble (right) with $k = 100$ and $\bar{d} = 5.87$. The precode is a shortened extended Hamming code with $k' = 92$.

Chapter 7

Unequally Loss-Resilient LT Code Ensembles

The first designs and studies of rateless codes targeted at applications like the loss-resilient distribution of bulk data [BLMR98]. Since in bulk data each bit is considered equally important, equal erasure protection (EEP) is the established mode for such transmissions. For other types of data, however, where some parts are more important than others and therefore need a stronger protection, unequal erasure protection (UEP) is often better suited.

UEP rateless codes are particularly interesting for audio and video transmissions which enable so-called graceful degradation, i.e. the data stream consists of different parts which constitute different layers, some of which are essential (e.g. base layer) while some are only refinement layers that enhance the resolution (spatial or temporal) or the quality provided by the base layer. If necessary, such refinement layers may be discarded without affecting the layers below. A well-known example of a video codec allowing graceful degradation is the Scalable Video Coding (SVC) [SMW07] extension of the H.264/AVC video coding standard [IT13]. In the domain of speech and audio coding, the standard G.729.1 [IT06, GRT08] offers hierarchical bit stream layers that enable to scale the acoustic bandwidth and the audio quality.

There exist essentially two different types of UEP LT code ensembles in the literature. In [RVF07] a UEP LT code construction method is presented which shall be referred to as the weighting (W-UEP) method while in [VSS+07, VSS+09, SVD+09, Sej09, VS12] the expanding window (EW-UEP) method is proposed. Both by their own means modify the column weights of the LT code matrix. In the graph representation this implies a modification of the input nodes' connectivity. The connectivity of an input node directly reflects its protection level, i.e. a high connectivity and thus a high column weight yields a higher protection level of the corresponding input node than a low connectivity or column weight.

In the following, the two methods will be briefly explained. At the same time the originally binary methods and their finite-length analysis under optimal erasure decoding will be directly generalised to higher order Galois fields. Due to a substantial weakness of the W-UEP method as described in [RVF07], a modification is proposed after introducing the original approach. The aforementioned deficiency arises from a non-linear operation in the construction process which was presumably introduced for its significant simplification of the finite length analysis of the such created codes but it yields only discrete and irregularly spaced protection levels. The W-UEP method, however, can be re-engineered such as to prevent the occurrence of this weakness. To this end biased sampling of the input symbols will be used in Section 7.2 which allows for continuous effective weights and thus for continuous protection levels.

7.1 Weighted UEP LT Code Ensembles

The first UEP construction method for LT code ensembles has been proposed in [RVF07]. This original weighted UEP (W-UEP) approach shall first be briefly reviewed in the following, before an enhanced version is discussed in the next section. The k input symbols are initially assigned to T importance classes. An importance class τ with $\tau \in \{1, \dots, T\}$ contains k_τ symbols, i.e. the size of class τ is given by $k_\tau = \alpha_\tau k$, $0 \leq \alpha_\tau \leq 1$ and $\sum_{i=1}^{T} \alpha_i = 1$, where α_τ is the relative size of importance class τ.

In contrast to the EEP construction, where the input nodes are chosen uniformly at random from the set of k input nodes to be connected to a given output node, i.e. with initial probability $p = p^{(1)} = \frac{1}{k}$, the W-UEP construction requires weighting factors φ_τ that are in accordance with the importance of the respective classes. The keyword *initial* implies connecting a new output node, which is yet unconnected, to the current graph by creating the first of d edges to a random input node. Accordingly, the probability of connecting a particular input node with the j^{th} edge, where $1 \leq j \leq d$, i.e. given that $j - 1$ input nodes are already connected, is denoted $p^{(j)}$. Consequently, the new initial probability of connecting a class τ input node to a given output node is $p_\tau = p_\tau^{(1)} = \varphi_\tau p = \frac{\varphi_\tau}{k}$. The weighting factors and the (relative) sizes have to fulfil the condition $\sum_{i=1}^{T} \varphi_i \alpha_i = \sum_{i=1}^{T} p_i k_i = 1$.

Presumably in order to facilitate the finite length analysis of these ensembles, the construction in [RVF07] has been confined such as to fix the number d_τ of input nodes from an arbitrary class τ that are to be connected to a particular output node of given degree d, i.e. d_τ is set to $\min(\text{rnd}(\alpha_\tau \varphi_\tau d),\, k_\tau)$. Therefore, this version of the weighted approach will hereafter be referred to as weighted UEP with rounded degrees. However, these means of fixing d_τ given d entail an inaccuracy in adjusting the protection levels or more directly the effective weights $\varphi_\tau^{[\text{eff}]}$, as the latter are discontinuous functions of the target weights φ_τ:

$$\varphi_\tau^{[\text{eff}]} = \frac{\bar{d}_\tau}{\alpha_\tau \bar{d}} = \frac{\sum_{d \in \mathcal{D}} \Omega_d \min(\text{rnd}(\alpha_\tau \varphi_\tau d),\, k_\tau)}{\alpha_\tau \bar{d}}. \tag{7.1}$$

The numerator in (7.1) is the effective average row weight $\bar{d}_\tau$ of class τ, while the denominator equals the average row weight that falls into that part of the LT code generator matrix which is associated with class τ in the case of EEP.

In Figure 7.1 the blue graphs show the effective weight $\varphi_1^{[\text{eff}]}$ as a function of the target weight φ_1 for the respective first of two importance classes with relative input sizes α_1 and α_2 for three exemplary weighted UEP LT code ensembles. The dashed black line indicates the optimal case $\varphi_1^{[\text{eff}]} = \varphi_1$. The main drawback of the rounded degrees approach should now become apparent, since the effective weights not only bend off the optimal course, but they do so in a discontinuous manner, due to which not all effective weights can be attained with a given parameter set, i.e. with given k, α_τ and $\Omega(\xi)$.

The first two ensembles have an input size of $k = 100$, but different row weight distributions, whereas the third ensemble has a larger input size of $k = 10000$ but the same row weight distribution as the second ensemble. As the effective weight $\varphi_\tau^{[\text{eff}]}$ is also a function of the overall row weight distribution $\Omega(\xi)$, the latter naturally affects the arising discontinuities. The effect of different row weight distributions becomes visible in the strongly differing step size irregularity when comparing Figures 7.1(a) and 7.1(b).

Contrasting the blue graphs in Figures 7.1(b) and 7.1(c) for different input sizes shows that one is roughly just a bent version of the other, i.e. for short codes and target weights greater than one the effective weight deviates towards lower values. This deviation is due to the operation $\min(\cdot)$ that limits the number of edges connected to the current importance class to the number of available input nodes. Clipping, however, only occurs for row weights d which are higher than the input sizes of the involved classes. Consequently, the operation $\min(\cdot)$ does not affect the long code in the depicted range of target weights.

7.2 Biased Sampling of Input Nodes

In order to allow for a continuous relation between φ_τ and $\varphi_\tau^{[\text{eff}]}$, a different method shall be used to select the input nodes of different classes to connect them to the current output node of degree d. The proposed method [SL12] uses biased sampling of the input nodes. Biased sampling is best explained by means of an urn model, where the differently important input nodes are represented by $k = \sum_{i=1}^{T} k_i$ balls of T different colours in an urn and the k_τ balls of a particular colour τ have an associated weight φ_τ. The urn experiment is carried out by drawing d balls one by one without replacement from such an urn, where the probability of picking a particular ball of a given colour τ at a particular draw j is proportional to the ball's relative weight with respect to the total weight of the current urn content:

$$p_\tau^{(j)} = \frac{\varphi_\tau}{\sum_{i=1}^{T} \varphi_i k_i^{(j-1)}}, \tag{7.2}$$

where $k_i^{(j-1)}$ denotes the number of balls of colour i in the urn at the end of draw $j-1$ and $k_i^{(0)} = k_i$ serves as initial number of balls of colour i in the urn. A detailed example is provided in Figure 7.2.

In [Wal63] Wallenius analysed biased sampling for the univariate case ($T = 2$). Chesson generalised the analysis to the multivariate case in [Che76]. Hence, the partitioning of the overall degree d of a specific output node into class degrees d_i, with $\sum_{i=1}^{T} d_i = d$, follows the so-called multivariate Wallenius' non-central hypergeometric distribution which denotes the conditional pmf

$$\Pr\Big\{\mathsf{d}_1 = d_1, \ldots, \mathsf{d}_{T-1} = d_{T-1} \Big| \mathsf{d} = d;\, \mathbf{k},\, \boldsymbol{\varphi}\Big\} = \left(\prod_{i=1}^{T} \binom{k_i}{d_i}\right) \int_0^1 \prod_{i=1}^{T} \Big(1 - t^{\frac{\varphi_i}{\boldsymbol{\varphi}(\mathbf{k}-\mathbf{d})}}\Big)^{d_i} \mathrm{d}t, \tag{7.3}$$

where $\mathbf{k} = \{k_1, k_2, \ldots, k_T\}$ comprises the class sizes and $\boldsymbol{\varphi} = \{\varphi_1, \varphi_2, \ldots, \varphi_T\}$ comprises the class specific target weights. The pmf in (7.3) can be determined by means of numerical integration as described in [Fog08b, Fog08a] using the BiasedUrn package [Fog11] for GNU R. To simplify the notation, the explicit parametrisation of the left hand side of (7.3) with $\mathbf{k}$ and $\boldsymbol{\varphi}$ will be omitted hereafter. Moreover, for a better legibility, an additional notation simplification will be used: given an arbitrary function $f(\mathbf{d})$, the collated sum

$$\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} f(\mathbf{d}) \qquad \text{denotes} \qquad \sum_{d_1} \cdots \sum_{d_T} f(\mathbf{d}),$$

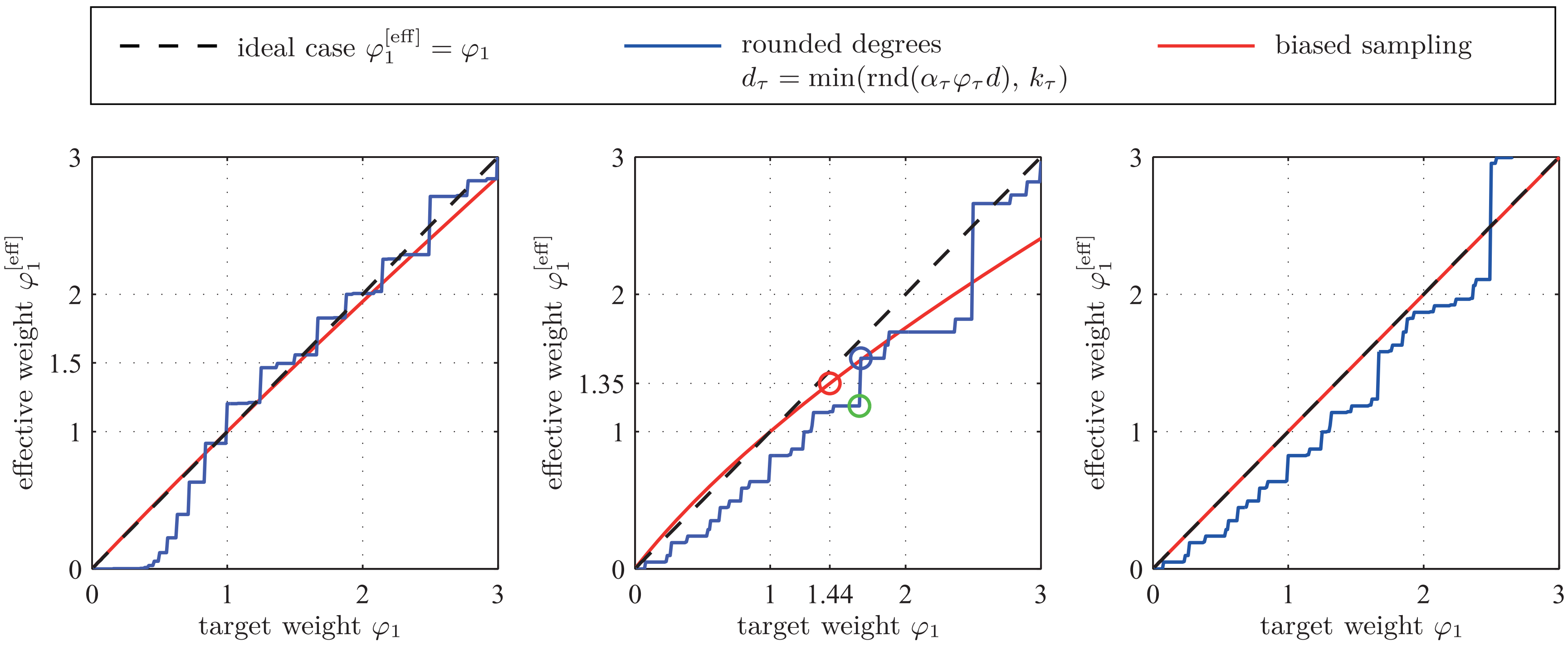
where the sums are calculated for all combinations of the values of $\mathbf{d} = (d_1, d_2, \ldots, d_T)^{\mathsf{T}}$ for which $1 \leq d \leq d_{\max}$ and $\sum_{\tau=1}^{T} d_\tau = d$. Additionally, $0 \leq d_\tau \leq \min(d, k_\tau)$ with $1 \leq \tau \leq T$.

Now, the multivariate row weight distribution

$$\begin{aligned} \boldsymbol{\Omega}(\boldsymbol{\xi}) &= \sum_{d_1} \cdots \sum_{d_T} \Omega_{d_1,\ldots,d_T} \xi_1^{d_1} \cdot \ldots \cdot \xi_T^{d_T} && (7.4) \\ &= \sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} \Omega_{\mathbf{d}} \boldsymbol{\xi}^{\mathbf{d}}, && (7.5) \end{aligned}$$

is required for determining upper and lower bounds on the residual erasure probabilities, where

$$\begin{aligned} \Omega_{\mathbf{d}} &= \Pr\{\boldsymbol{\mathsf{d}} = \mathbf{d}\} = \Pr\{\mathsf{d}_1 = d_1, \ldots, \mathsf{d}_T = d_T\} \\ &= \Pr\{\mathsf{d}_1 = d_1, \ldots, \mathsf{d}_{T-1} = d_{T-1},\, \mathsf{d} = d\} \\ &= \Pr\{\mathsf{d} = d\} \cdot \Pr\Big\{\mathsf{d}_1 = d_1, \ldots, \mathsf{d}_{T-1} = d_{T-1} \Big| \mathsf{d} = d\Big\} \\ &= \Omega_d \cdot \Pr\Big\{\mathsf{d}_1 = d_1, \ldots, \mathsf{d}_{T-1} = d_{T-1} \Big| \mathsf{d} = d\Big\}. && (7.6) \end{aligned}$$

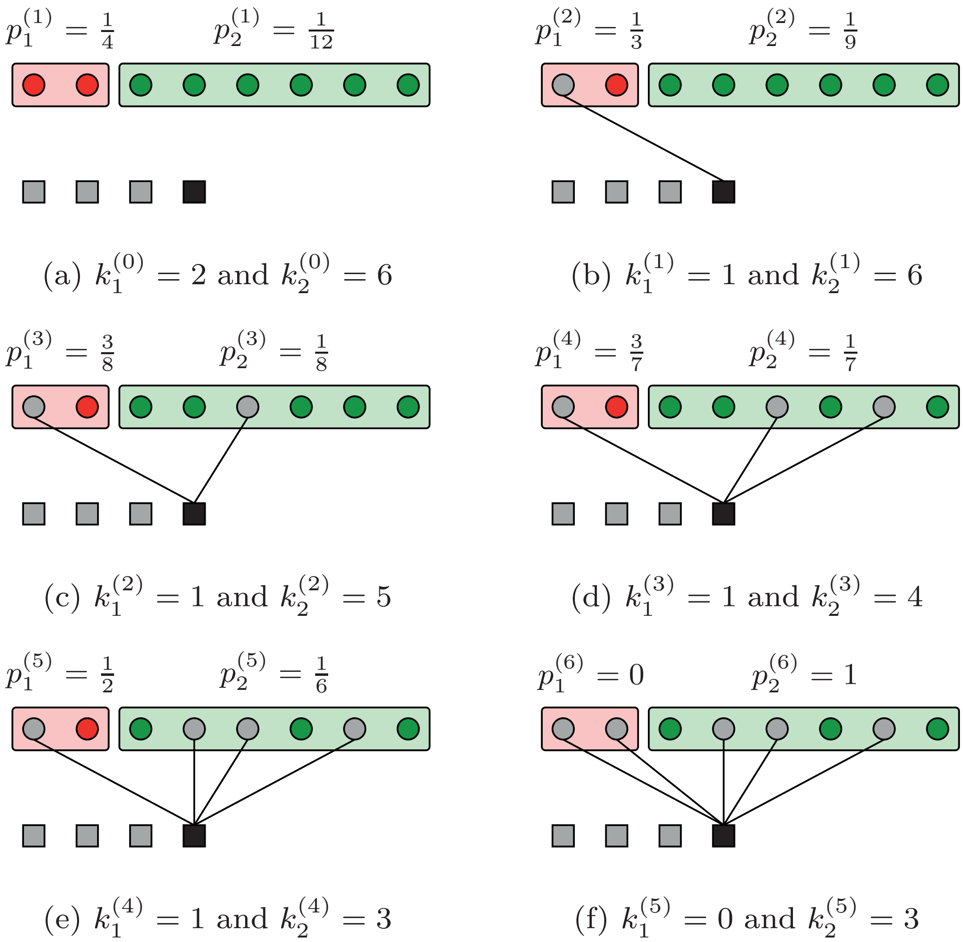


(a) Sparse random ensemble, $k = 100$ and $\bar{d} = 5.87$.

(b) BP-optimised ensemble, $k = 100$ and $\bar{d} = 5.87$.

(c) BP-optimised ensemble, $k = 10000$ and $\bar{d} = 5.87$.

Figure 7.1: The effective weight $\varphi_1^{[\mathrm{eff}]}$ of class 1 as a function of the target weight φ_1 for three weighted UEP LT code ensembles with two classes of relative sizes $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. The ensembles in (a) and (b) have an input size of $k = 100$, while the one in (c) has an input size of 10000. In (b) and (c) the BP-optimised overall row weight distribution (2.8) has been used which yields an average row weight $\bar{d} = 5.87$. The round markers in (b) are operating points used for Figure 7.3.


$p_1^{(1)} = \frac{1}{4}$
$p_2^{(1)} = \frac{1}{12}$
$p_1^{(2)} = \frac{1}{3}$
$p_2^{(2)} = \frac{1}{9}$
$p_1^{(3)} = \frac{3}{8}$
$p_2^{(3)} = \frac{1}{8}$
$p_1^{(4)} = \frac{3}{7}$
$p_2^{(4)} = \frac{1}{7}$
$p_1^{(5)} = \frac{1}{2}$
$p_2^{(5)} = \frac{1}{6}$
$p_1^{(6)} = 0$
$p_2^{(6)} = 1$

(a) $k_1^{(0)} = 2$ and $k_2^{(0)} = 6$

(b) $k_1^{(1)} = 1$ and $k_2^{(1)} = 6$

(c) $k_1^{(2)} = 1$ and $k_2^{(2)} = 5$

(d) $k_1^{(3)} = 1$ and $k_2^{(3)} = 4$

(e) $k_1^{(4)} = 1$ and $k_2^{(4)} = 3$

(f) $k_1^{(5)} = 0$ and $k_2^{(5)} = 3$

Figure 7.2: One possible realisation of connecting input nodes to the current output node (black square) of degree $d = 5$ via biased sampling given an input size of $k = 8$, $T = 2$ importance classes of relative sizes $\alpha_1 = 0.25$ and $\alpha_2 = 0.75$ and class weights $\varphi_1 = 2$ and $\varphi_2 = \frac{2}{3}$. After each of the d draws, enumerated by j, $1 \leq j \leq d$, the probabilities $p_1^{(j)}$ and $p_2^{(j)}$ of connecting the j^{th} edge to a specific input node of class 1 or 2 have to be updated, taking into account the remaining $k_1^{(j-1)}$ and $k_2^{(j-1)}$ unconnected input nodes in either class, i.e. $p_1^{(j)} = \frac{\varphi_1}{\varphi_1 k_1^{(j-1)} + \varphi_2 k_2^{(j-1)}}$ and $p_2^{(j)} = \frac{\varphi_2}{\varphi_1 k_1^{(j-1)} + \varphi_2 k_2^{(j-1)}}$.

The second term in (7.6) is the multivariate Wallenius' non-central hypergeometric distribution (7.3). Accordingly, the average row weight associated with class τ is

$$\bar{d}_\tau = \sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} \Omega_{\mathbf{d}} \frac{d_\tau}{k_\tau} \tag{7.7}$$

and thus, the effective weight can be determined as

$$\varphi_\tau^{[\text{eff}]} = \frac{\bar{d}_\tau}{\alpha_\tau \bar{d}} = \frac{\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} \Omega_{\mathbf{d}} \frac{d_\tau}{k_\tau}}{\alpha_\tau \bar{d}}. \tag{7.8}$$

Like for the original weighted UEP approach above, the effective weight for biased sampling is depicted in red in Figure 7.1 for the same exemplary LT code ensembles. The effect of the modification becomes apparent in that the effective weight $\varphi_\tau^{[\mathrm{eff}]}$ is now a continuous function of the target weight φ_τ. So despite the deviation from the ideal case $\varphi_\tau^{[\mathrm{eff}]} = \varphi_\tau$ for the short ensembles, any desired effective weight can be attained by the biased sampling approach.

7.2.1 Finite Length Analysis

As in the EEP case, the erasure correction performance of UEP LT code ensembles can be assessed by upper and lower bounds on word and symbol level for any importance class τ. The proofs of the following corollaries are similar to the ones of Theorems 3.14, 3.15, 3.17 and 3.18 and they are provided in Appendix A.

Corollary 7.1. Given a weighted UEP LT code ensemble $\mathfrak{L}$ over $\mathbb{F}_q$ using biased sampling, an upper bound on the word erasure probability $P_\tau^{[\mathfrak{L}]}\big(\cancel{W}\big)$ of importance class τ after ML decoding is

$$\overline{P}_\tau^{[\mathfrak{L}]}\big(\cancel{W}\big) = \sum_{\substack{w=w_1+\ldots+w_T=1\\ w_\tau\geq 1}}^{k} (q-1)^{w-1}\left(\prod_{i=1}^{T}\binom{k_i}{w_i}\right) \cdot \left[\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} \Omega_{\mathbf{d}} \sum_{l=l_1+\ldots+l_T=0}^{d} \frac{1}{q}\left(1-(1-q)^{1-l}\right)\prod_{i=1}^{T}\frac{\binom{w_i}{l_i}\binom{k_i-w_i}{d_i-l_i}}{\binom{k_i}{d_i}}\right]^{k\gamma_{\mathrm{R}}}. \quad (7.9)$$

Corollary 7.2 (from [SL12])**.** Accordingly, an upper bound on the symbol erasure probability $P_\tau^{[\mathfrak{L}]}\big(\cancel{S}\big)$ of importance class τ after ML decoding is

$$\overline{P}_\tau^{[\mathfrak{L}]}\big(\cancel{S}\big) = \sum_{\substack{w=w_1+\ldots+w_T=1\\ w_\tau\geq 1}}^{k-1} (q-1)^{w-1}\left(\prod_{i=1}^{T}\binom{k_i-\delta_{i,\tau}}{w_i-\delta_{i,\tau}}\right) \cdot \left[\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} \Omega_{\mathbf{d}} \sum_{l=l_1+\ldots+l_T=0}^{d} \frac{1}{q}\left(1-(1-q)^{1-l}\right)\prod_{i=1}^{T}\frac{\binom{w_i}{l_i}\binom{k_i-w_i}{d_i-l_i}}{\binom{k_i}{d_i}}\right]^{k\gamma_{\mathrm{R}}}, \quad (7.10)$$

where $\delta_{i,j}$ is the Kronecker delta function, which equals one if $i = j$ and zero otherwise.

Corollary 7.3. A lower bound on the word erasure probability $P_\tau^{[\mathfrak{L}]}\big(\cancel{W}\big)$ of importance class τ after ML decoding is

$$\underline{P}_\tau^{[\mathfrak{L}]}\big(\cancel{W}\big) = \sum_{i=1}^{k_\tau}(-1)^{i+1}\binom{k_\tau}{i}\left(\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} \Omega_{\mathbf{d}}\frac{\binom{k_\tau-i}{d_\tau}}{\binom{k_\tau}{d_\tau}}\right)^{k\gamma_{\mathrm{R}}}. \quad (7.11)$$

Corollary 7.4 (from [SL12]). And finally, a lower bound on the symbol erasure probability $P_\tau^{[\mathfrak{L}]}\big(\not S\big)$ of importance class τ after ML decoding is given by

$$\underline{P}_\tau^{[\mathfrak{L}]}\big(\not S\big) = \left(1 - \sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} \Omega_{\mathbf{d}} \frac{d_\tau}{k_\tau}\right)^{k\gamma_{\mathrm{R}}} . \tag{7.12}$$

7.2.2 Numerical Evaluation and Monte Carlo Simulations

In order to further illustrate the difference between the original weighted UEP approach and the one with biased sampling of input nodes, three exemplary weighted UEP LT code ensembles, each with two importance classes, are to be constructed to achieve the effective weight $\varphi_1^{[\mathrm{eff}]} = 1.35$, i.e. importance class 1 obtains a stronger protection than class 2. The input size is $k = 100$, the relative class sizes are $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$, and the used row weight distribution is the BP-optimised one given by (2.8), i.e. the three ensembles are the same as the ones used in Figure 7.1(b) with the weights as indicated by the round markers. In Figure 7.3, bounds on the residual symbol erasure probability $P_\tau^{[\mathfrak{L}]}\big(\not S\big)$ are depicted for the three ensembles and the two respective importance classes.

With the original weighted UEP method the effective weight $\varphi_1^{[\mathrm{eff}]} = 1.35$ cannot be attained with the given code parameters. The closest approach is achieved by using the target weights left or right of the weight discontinuity, i.e. either $\varphi_1 = 1.66$ ($\varphi_1^{[\mathrm{eff}]} = 1.187$, green circle) or $\varphi_1 = 1.67$ ($\varphi_1^{[\mathrm{eff}]} = 1.534$, blue circle), which lead to the bounds depicted in green or blue, respectively. In one case (blue), class 1 is protected too well at the cost of class 2, while in the other case (green), class 1 is not protected sufficiently. In general, the protection of a smaller class is usually more influenced than that of a larger class and the effective protection levels cannot be accurately adjusted using the original method. By employing biased sampling, however, any effective weight and thus any protection level can be obtained. For the present example this is achieved by setting $\varphi_1 = 1.44$ (red curves).

7.2.3 Practical Design of Sparse Random UEP Ensembles

Sparse random ensembles not only show a remarkable performance in EEP scenarios but also with UEP constraints. Like in the case of EEP, the performance of UEP sparse random ensembles can be easily assessed by means of the derived bounds on the residual erasure probabilities. The computation of the bounds in (7.9) to (7.12) (particularly the upper bounds), however, can be quite time-consuming if the input size is large ($k \gg 100$), if the number of importance classes is greater than 3 or 4 or if the row weight distribution is not sparse, i.e. if the cardinality of the row weight sample space $|\mathcal{D}|$ is large. So for UEP ensembles that are based on sparse random ensembles, $|\mathcal{D}|$ is equal to the input size and is thus not small.

—— upper bounds – – – lower bounds

ideal	🟥 biased sampling	🟦 rounded degrees	🟩 rounded degrees
$\varphi_1^{[\mathrm{eff}]} = 1.35$	$\varphi_1^{[\mathrm{eff}]} = 1.35$	$\varphi_1^{[\mathrm{eff}]} = 1.534$	$\varphi_1^{[\mathrm{eff}]} = 1.187$
$\varphi_1 = 1.35$	$\varphi_1 = 1.44$	$\varphi_1 = 1.67$	$\varphi_1 = 1.66$
$\varphi_2^{[\mathrm{eff}]} = 0.961$	$\varphi_2^{[\mathrm{eff}]} = 0.961$	$\varphi_2^{[\mathrm{eff}]} = 0.935$	$\varphi_2^{[\mathrm{eff}]} = 0.974$
$\varphi_2 = 0.961$	$\varphi_2 = 0.951$	$\varphi_2 = 0.9256$	$\varphi_2 = 0.9267$

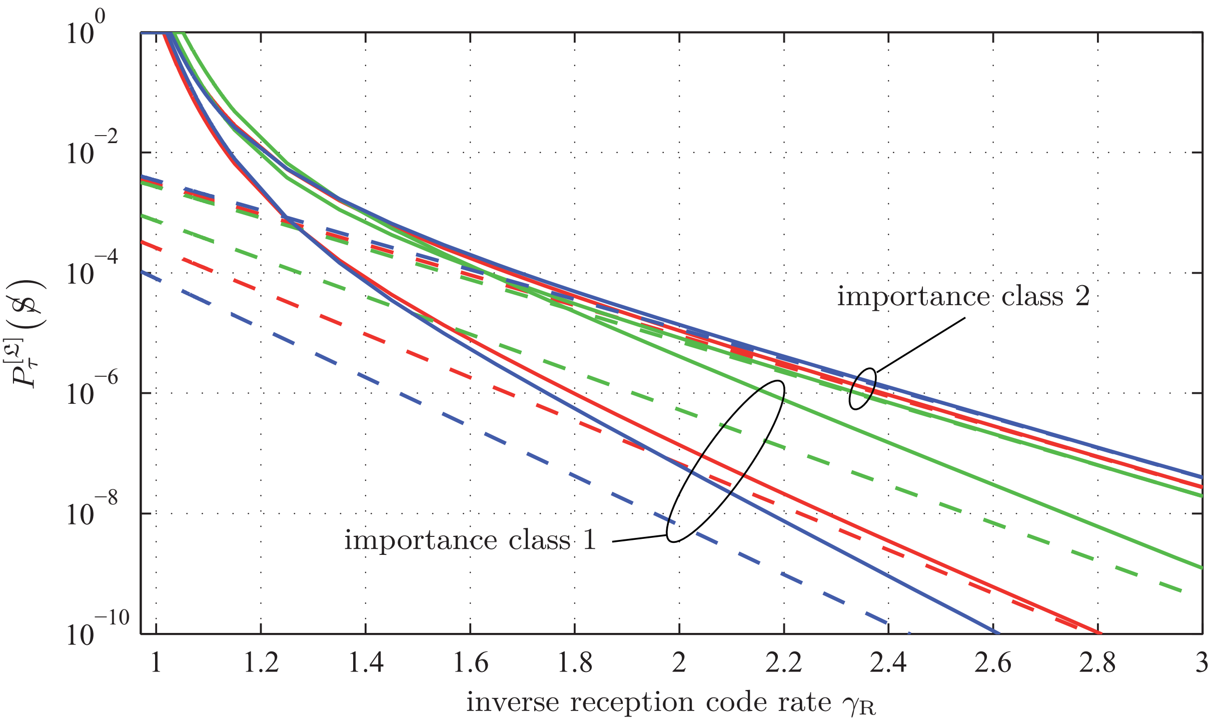


Figure 7.3: Upper and lower bounds for three UEP LT code ensembles with $k = 100$, two importance classes and relative class sizes of 0.1 and 0.9. An effective weight $\varphi_1^{[\mathrm{eff}]} = 1.35$ shall be obtained. The weights φ_1 that yield the closest values to $\varphi_1^{[\mathrm{eff}]}$ for the different code construction methods are highlighted in Figure 7.1(b) by round markers with corresponding colours. Using the rounded degrees method, the envisaged $\varphi_1^{[\mathrm{eff}]} = 1.35$ cannot be reached. The closest one can get is by using either $\varphi_1 = 1.66$ ($\varphi_1^{[\mathrm{eff}]} = 1.187$, green curves) or $\varphi_1 = 1.67$ ($\varphi_1^{[\mathrm{eff}]} = 1.534$, blue curves). By biased sampling on the other hand, any effective weight $\varphi_1^{[\mathrm{eff}]}$ and thus any protection level can be attained.

This means that particularly for such ensembles the computation of the bounds, one of the first steps in the design process, can become lengthy.

One possibility to speed up the process is to set the multitude of extremely small values Ω_d to zero and thereby to decrease $|\mathcal{D}|$ without affecting the accuracy. Another possibility is to use the bounds of *equivalent* EEP sparse random ensembles,

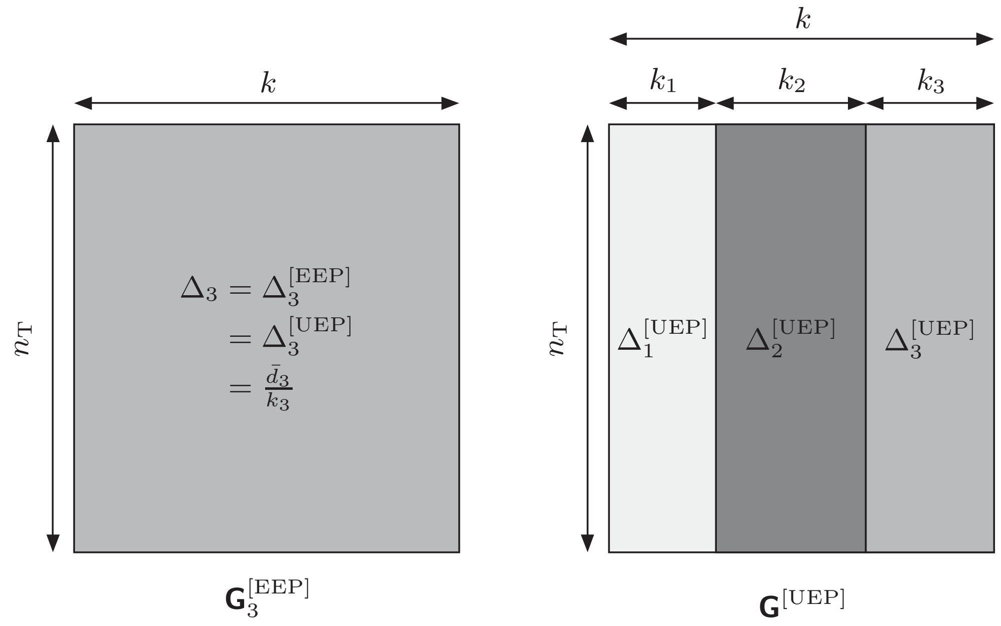

k
n_{T}
$\Delta_3 = \Delta_3^{[\mathrm{EEP}]}$
$= \Delta_3^{[\mathrm{UEP}]}$
$= \frac{\bar{d}_3}{k_3}$
$\mathbf{G}_3^{[\mathrm{EEP}]}$
k
k_1
k_2
k_3
n_{T}
$\Delta_1^{[\mathrm{UEP}]}$
$\Delta_2^{[\mathrm{UEP}]}$
$\Delta_3^{[\mathrm{UEP}]}$
$\mathbf{G}^{[\mathrm{UEP}]}$

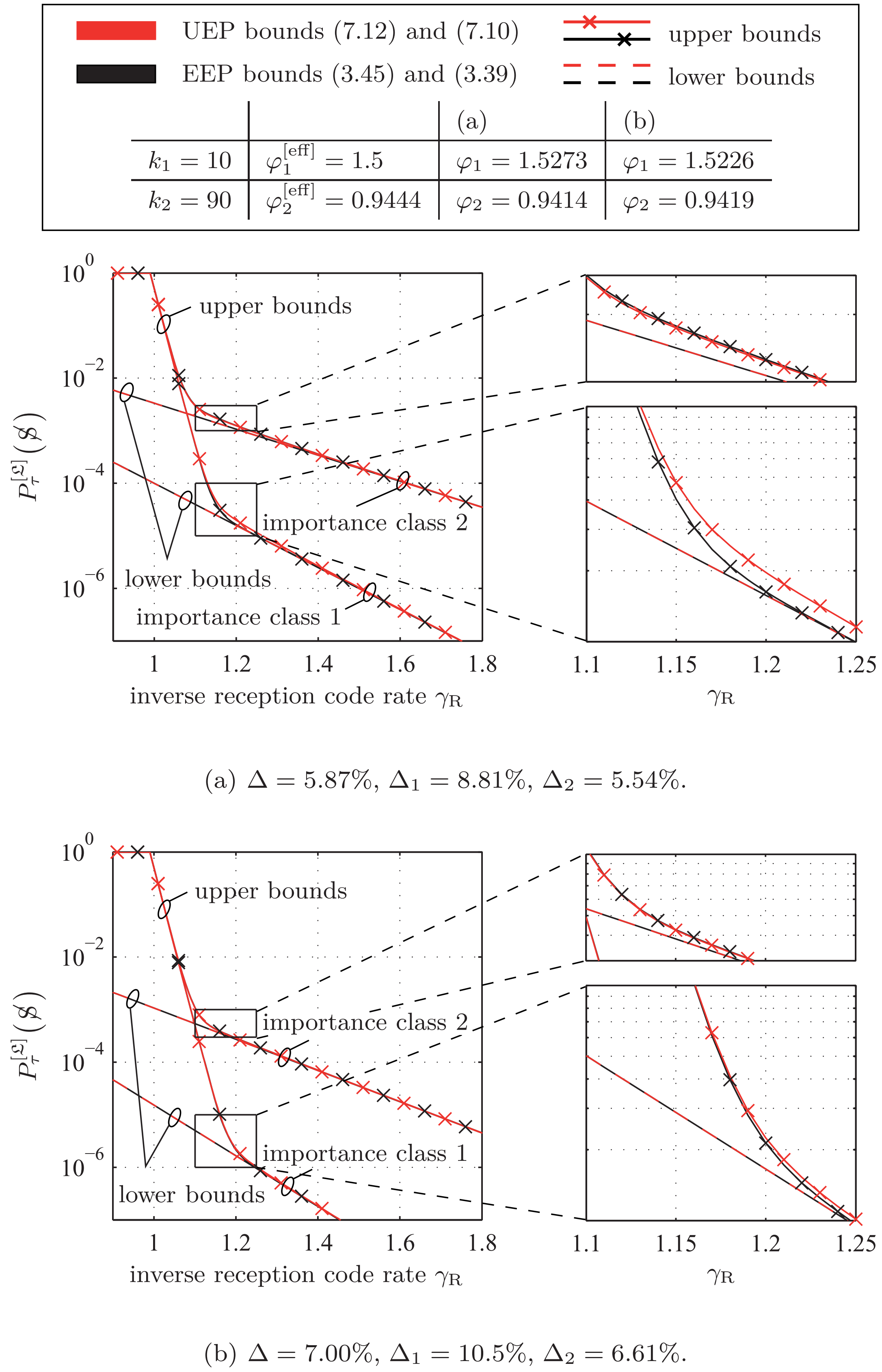
Figure 7.4: Illustration of an equivalent EEP sparse random LT code ensemble that can be used to approximate the bounds of the third importance class of a UEP sparse random LT code ensemble which is constructed by means of biased sampling.

which can be computed very fast [SV12]. What exactly is understood by the term “equivalent” will be explained subsequently. Note, however, that the approximation of the UEP bounds with the equivalent EEP bounds is only feasible for sparse random ensembles, not for UEP LT code ensembles in general.

Sparse random LT code ensembles, in contrast to other LT code ensembles, are parametrised with the average row weight $\bar{d}$ (alternatively the density Δ) and the input size k. By biased sampling, the entries in the LT code matrix are distributed such that in the different parts of the matrix which correspond to the respective importance classes, the sparse random property is conserved when compared with an EEP sparse random matrix of the same overall density. The individual densities, however, which correspond to the respective protection levels normally differ from the previous EEP density by implication.

Now, evaluating the performance for class τ reveals that it is almost identical to the performance of another EEP sparse random ensemble whose density $\Delta_\tau = \Delta_\tau^{[\mathrm{EEP}]}$ is equal to the density $\Delta_\tau^{[\mathrm{UEP}]}$ in the respective part of the UEP matrix, i.e. $\Delta_\tau^{[\mathrm{EEP}]} = \Delta_\tau^{[\mathrm{UEP}]} = \frac{\bar{d}_\tau}{k_\tau}$. Moreover, the EEP matrix and the UEP matrix are supposed to have the same width k. Such an EEP ensemble is considered to be equivalent to a particular importance class τ and therefore allows to approximate the bounds of this importance class. An illustration is provided in Figure 7.4.

In Figure 7.5 upper and lower bounds on the residual symbol erasure probability $P_\tau^{[\mathfrak{L}]}\left(\not{\$}\right)$ are drawn in red for two exemplary sparse random UEP ensembles with



		(a)	(b)
$k_1 = 10$	$\varphi_1^{[\mathrm{eff}]} = 1.5$	$\varphi_1 = 1.5273$	$\varphi_1 = 1.5226$
$k_2 = 90$	$\varphi_2^{[\mathrm{eff}]} = 0.9444$	$\varphi_2 = 0.9414$	$\varphi_2 = 0.9419$

(a) $\Delta = 5.87\%$, $\Delta_1 = 8.81\%$, $\Delta_2 = 5.54\%$.

(b) $\Delta = 7.00\%$, $\Delta_1 = 10.5\%$, $\Delta_2 = 6.61\%$.

Figure 7.5: Upper and lower bounds on the symbol erasure probability $P_\tau^{[\mathfrak{L}]}(\not{S})$ for two binary UEP LT code ensembles as well as the bounds of the sparse random EEP LT code ensembles that correspond to the UEP importance classes.

two importance classes and an overall input size of $k = 100$. The ensemble used for Figure 7.5(a) has an overall density of $\Delta = 5.87\%$, while the overall density of the ensemble used in Figure 7.5(b) is $\Delta = 7.00\%$. The set of used parameters is summarised in the table provided in Figure 7.5 as well as below the subfigures. Additionally, the bounds of the equivalent EEP sparse random ensembles are depicted in black.

The upper bounds represent approximations to the respective UEP upper bounds, whereas the corresponding EEP and UEP lower bounds are equal. Due to the strong congruence of the EEP and the UEP upper bounds, the transitional parts between waterfall and erasure floor are magnified, particularly since the divergence of the approximations to the true bounds is maximal though small in this area. In general it can be stated that the accuracy improves with a higher overall density. Moreover, the deviation of the bounds for larger importance classes is lower than for smaller importance classes. Keeping these effects in mind when using the approximations, the initial design steps based on bound computation for sparse random UEP LT ensembles can be performed with an accuracy and speed comparable to that of EEP ensembles.

7.3 Expanding Window LT Code Ensembles

The expanding window (EW) method [SVD$^+$09, Sej09] is another way to impose different protection levels on different parts of an information word. To this end, w.l.o.g. the symbols in the information word $\mathbf{x}$ are considered to be ordered with respect to their importance as sketched in Figure 7.6, where the EW construction process is depicted. Let T denote the number of different importance classes or protection levels, then $\mathbf{x}_\tau$ contains the k_τ information symbols in importance class $\tau \in \{1, \ldots, T\}$, where $\tau = 1$ labels the most important class. The complete information word is thus $\mathbf{x} = \left(\mathbf{x}_1^\mathsf{T}, \ldots, \mathbf{x}_T^\mathsf{T}\right)^\mathsf{T}$.

Now, the usual LT code construction method is supplemented with some additional steps. The importance classes are assigned to T windows, such that the first window comprises only the information symbols $\mathbf{x}_1$ of the most important class 1, the second window contains $\left(\mathbf{x}_1^\mathsf{T}, \mathbf{x}_2^\mathsf{T}\right)^\mathsf{T}$ and the τ^{th} window consists of $\left(\mathbf{x}_1^\mathsf{T}, \ldots, \mathbf{x}_\tau^\mathsf{T}\right)^\mathsf{T}$. In this manner, the more important classes are included in many windows, while less important classes are contained only in few windows.

When generating an encoded symbol, i.e. a row in the LT code matrix, it is first assigned randomly to a window according to a window selection distribution $\mathcal{W}(\xi) = \sum_{\tau=1}^{T} \mathcal{W}_\tau \xi^\tau$. Furthermore, each window τ has an individual row weight distribution $\Omega_\tau(\xi) = \sum_{d=1}^{\kappa_\tau} \Omega_{\tau,d} \xi^d$ from which the current row weight d is sampled, where $\kappa_\tau = \sum_{i=1}^{\tau} k_i$ is the number of input symbols in the τ^{th} window. Finally, d input symbols from the current window are chosen uniformly at random

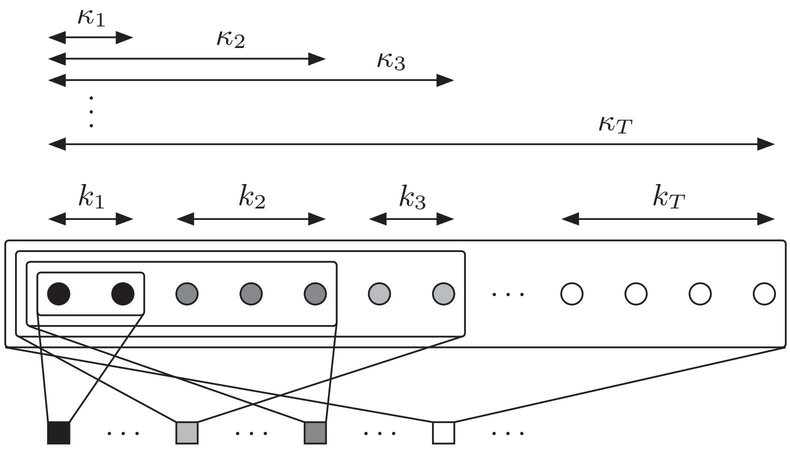


Figure 7.6: Illustration of the EW code construction: the input nodes are assigned to T importance classes of size k_τ, $\tau \in \{1, \ldots, T\}$ in descending order of importance. Windowing is performed such that the τ^{th} window contains importance classes 1 to τ. An output node is created by first assigning it to a window τ according to the window selection distribution $\mathcal{W}(\xi)$, second by sampling a degree d from the row weight distribution $\Omega_\tau(\xi)$ and finally by connecting it to d input nodes chosen uniformly at random from the τ^{th} window.

to create the respective encoded symbol, i.e. the current row in the LT code matrix has d non-zero entries within the leftmost κ_τ columns. As usual, each of the d non-zero entries is chosen uniformly at random from $\mathbb{F}_q \setminus \{0\}$.

The EW construction allows two degrees of freedom to achieve UEP. By increasing (decreasing) the average row weight in the more (less) important part of the LT code matrix, the erasure floor is lowered (heightened). Moreover, by increasing the probability of choosing smaller windows (which contain the more important symbols) the waterfall can be reached earlier for these symbols at the cost of a later waterfall for the less important ones.

7.3.1 Finite Length Analysis

The following four corollaries provide upper and lower bounds on the residual erasure probability on word and on symbol level after optimal erasure decoding for a fixed but arbitrary importance class τ. The proofs of the following corollaries are similar to the ones of Theorems 3.14, 3.15, 3.17 and 3.18 as well as similar to the proofs of the respective bounds in the case of biased sampling and they are provided in Appendix A.

Corollary 7.5. Given an EW-UEP LT code ensemble $\mathfrak{L}$ over $\mathbb{F}_q$ with T different protection levels, an upper bound on the word erasure probability $P_\tau^{[\mathfrak{L}]}\big(\cancel{W}\big)$ of importance class $\tau \in \{1, \ldots, T\}$ after ML decoding is

$$\overline{P}_\tau^{[\mathfrak{L}]}\big(\not{W}\big) = \sum_{w_T=1}^{\kappa_T} \cdots \sum_{w_\tau=1}^{\kappa_\tau} \sum_{w_{\tau-1}=0}^{\kappa_{\tau-1}} \cdots \sum_{w_1=0}^{\kappa_1} (q-1)^{w_T-1} \left(\prod_{i=1}^{T} \binom{k_i}{w_i - w_{i-1}} \right) \cdot \left[\frac{1}{q} \sum_{j=1}^{T} \mathcal{W}_j \sum_{d=1}^{\kappa_j} \Omega_{j,d} \frac{\sum_{l=0}^{d} \binom{w_j}{l} \binom{\kappa_j - w_j}{d-l} \left[1 - (1-q)^{1-l}\right]}{\binom{\kappa_j}{d}} \right]^{k\gamma_{\mathrm{R}}} , \quad (7.13)$$

with the inverse reception code rate $\gamma_{\mathrm{R}} = 1 + \varepsilon_{\mathrm{R}}$, the window specific row weight distributions $\Omega_i(\xi)$, $i \in \{1, \ldots, T\}$, the window selection distribution $\mathcal{W}(\xi)$ and $w_0 = 0$.

Corollary 7.6. A corresponding upper bound on the symbol erasure probability $P_\tau^{[\mathfrak{L}]}\big(\not{S}\big)$ of importance class τ after ML decoding is then

$$\overline{P}_\tau^{[\mathfrak{L}]}\big(\not{S}\big) = \sum_{w_T=1}^{\kappa_T} \cdots \sum_{w_\tau=1}^{\kappa_\tau} \sum_{w_{\tau-1}=0}^{\kappa_{\tau-1}} \cdots \sum_{w_1=0}^{\kappa_1} (q-1)^{w_T-1} \left(\prod_{i=1}^{T} \binom{k_i - \delta_{i,\tau}}{w_i - w_{i-1} - \delta_{i,\tau}} \right) \cdot \left[\frac{1}{q} \sum_{j=1}^{T} \mathcal{W}_j \sum_{d=1}^{\kappa_j} \Omega_{j,d} \frac{\sum_{l=0}^{d} \binom{w_j}{l} \binom{\kappa_j - w_j}{d-l} \left[1 - (1-q)^{1-l}\right]}{\binom{\kappa_j}{d}} \right]^{k\gamma_{\mathrm{R}}} . \quad (7.14)$$

Corollary 7.6 is a generalisation of the upper bound on symbol level for the binary case from [SVD^{+}09, Sej09].

Corollary 7.7. A lower bound on the word erasure probability $P_\tau^{[\mathfrak{L}]}\big(\not{W}\big)$ of importance class τ after ML decoding is

$$\underline{P}_\tau^{[\mathfrak{L}]}\big(\not{W}\big) = \sum_{i=1}^{k_\tau} (-1)^{i+1} \binom{k_\tau}{i} \left(\sum_{j=\tau}^{T} \mathcal{W}_j \sum_{d=1}^{\kappa_\tau} \Omega_{j,d} \sum_{d_\tau=0}^{k_\tau} \frac{\binom{k_\tau - i}{d_\tau} \binom{\kappa_j - k_\tau}{d - d_\tau}}{\binom{\kappa_j}{d}} \right)^{k\gamma_{\mathrm{R}}} . \quad (7.15)$$

Corollary 7.8 (from [Sej09])**.** And finally, a lower bound on the symbol erasure probability $P_\tau^{[\mathfrak{L}]}\big(\not{S}\big)$ of importance class τ after ML decoding is

$$\underline{P}_\tau^{[\mathfrak{L}]}\big(\not{S}\big) = \left(1 - \sum_{i=\tau}^{T} \mathcal{W}_i \frac{\bar{d}_i}{k_i} \right)^{k\gamma_{\mathrm{R}}} . \quad (7.16)$$

7.4 Conclusions

Two UEP LT code construction methods, i.e. the weighting and the expanding window method, have been discussed in this chapter. The existing bounds on symbol level under optimal erasure decoding have been generalised to higher order Galois fields and new bounds on word level have been derived. Moreover, the accuracy of the weighting method has been improved by employing biased sampling of the input nodes for which the respective bounds have been provided as well. This modification facilitates continuous effective weights and consequently allows to attain any desired protection level. For the class of sparse random UEP LT code ensembles a simplified and less complex heuristic design method has been described. This method is based on the equivalent appearance and properties of the parts of the UEP LT code matrix that correspond to the respective importance classes and EEP LT code matrices with appropriate densities.

What has been left out is a generalised approach, i.e. a combination of the two methods, which allows windowing and weighting (based on biased sampling) at the same time. The finite length analysis of this combination, though straightforward, is rather lengthy and tedious without offering much further insight into the topic and is thus omitted here.

As a final subject, UEP Raptor codes remain to be addressed: here, the task of providing unequal protection is best deferred from the LT code to the precode. Particularly irregular LDPC codes seem to be highly eligible UEP precodes [RF04, RPNF07], which share many properties with the discussed UEP LT code ensembles. In fact, irregular LDPC codes, being a well-studied class of codes and being considered superior to regular ones [LMS$^+$97], naturally provide UEP.

Chapter 8

Summary

Since the first, merely theoretical, postulation of the digital fountain principle in [BLMR98], the progress in the development of practical fountain codes has been remarkable. So they have quickly found their way into numerous communication standards like for instance IETF RFCs 5053 and 6330 [LSWS07, LSW$^+$11], the 3GPP MBMS standard [3GP13] for multimedia broadcasting and multicasting services, the DVB-IPDC standard [ETS09b] for IP datacast over digital video broadcasting networks or the DVB-IPTV standard [ETS09a] for TV services over IP networks.

Digital fountain codes have two key properties which render them invaluable for erasure-resilient data transmissions. The first is their ratelessness which allows to create a potentially infinite number n of encoded packets from a finite number k of data packets. The second one is their near-MDS property which allows to recover the original k data packets from any $(1+\varepsilon)k$ out of the n encoded packets with a small overhead $\varepsilon \geq 0$.

These two properties facilitate a (near) optimal erasure-resilience that is independent from the channel conditions, which is denoted as universality. As the average contribution of each packet to the decodability of the original information is equal and (near) optimal, it does not make sense to request a retransmission of a particular lost packet, since any of the following packets is as useful as the lost one. This redundantises the need for acknowledging the reception of individual packets and thus the need for further communication between transmitter and receiver, besides an optional final feedback message upon successful completion of the overall recovery. Thus, provided that the IP network offers a multicast service, a single server that hosts popular content can serve a huge number of users simultaneously, even if they do not tune in at the same time. Moreover, the network traffic is reduced tremendously, since the same packet is sent to all subscribed users at a fixed but arbitrary time instant and no further signalling or retransmissions are required.

In a nutshell, the major contribution of this thesis is the thorough examination under various constraints of the erasure correction properties of two practically relevant types of rateless code ensembles over finite fields under optimal erasure decoding, i.e. Luby Transform (LT) codes and Raptor codes. Due to several practical advantages of non-binary codes over binary ones, the results in this thesis have been almost entirely presented in general form for arbitrary Galois field orders.

Though optimal erasure decoding is widely considered too complex for practical purposes, for the targeted short input sizes, which are relevant, e.g. for low-delay applications, it becomes affordable complexity-wise and yet almost mandatory, since suboptimal but less complex greedy algorithms do not achieve sufficiently low residual erasure rates. As optimal decoding of an LT code is equivalent to solving a consistent system of linear equations, where the coefficients are given by a pruned instance of an LT code generator matrix, which is created according to some specifically designed random processes, the achieved results may be transferable to other research fields where specifically designed systems of random linear equations also need to be evaluated for their solvability.

Finite Length Analysis under Optimal Erasure Decoding

Starting with the most random LT code ensemble, which is perhaps the best understood ensemble, i.e. the standard random ensemble, and the closely linked expurgated random ensemble, residual erasure probabilities after optimal decoding have been provided on word level. For the standard random ensemble, the respective probability is well known in the literature, while for the expurgated random ensemble, the optimal ensemble under optimal decoding, a new recursive expression has been derived. It is furthermore shown, that except for extremely small input sizes, the two ensembles perform indistinguishably well, so that for practical input sizes no distinction is necessary between the ensembles.

Since exact residual erasure probabilities are not known for general LT code ensembles but for the above-mentioned special cases, bounds thereon have been derived or existing ones for binary ensembles have been generalised for non-binary ones. The obtained set of four bounds, i.e. upper and lower bounds on symbol and on word level, has then been used to identify quasi-optimal ensembles with a density constraint such as the expurgated sparse random ensembles.

For low reception overheads, in the so-called waterfall area, sparse random ensembles have been shown to achieve almost the same erasure correction performance as their dense counterparts at a much lower computational cost. By adjusting the density of sparse random ensembles, it is extremely simple to trade off erasure resilience against computational complexity.

The upper bounds have turned out to be very good approximations of the true erasure rates, particularly for ensembles that show a good erasure correction performance under optimal decoding. They enable a quick assessment of the erasure correction properties of an ensemble without the need for time-consuming Monte Carlo simulations.

By moderately increasing the Galois field size and adjusting the ensemble parameters in a fair manner, it has been demonstrated that the erasure resilience is improved, while the computational complexity is lowered considerably at the same time.

Conventionally Systematic LT Code Ensembles

It is a widely held belief that conventionally systematic LT code ensembles, i.e. ensembles where an identity matrix is prepended to an ordinary LT code generator matrix, are inferior to ensembles created by the systematic construction. The set of four bounds on the residual erasure probability for ordinary LT code ensembles has been extended to conventionally systematic ensembles, permitting an accurate analysis of such ensembles. So it has been shown that, though the aforementioned inferiority remains undoubted in many cases, for some ensembles a conventionally systematic prefix is indeed beneficial, both with respect to their erasure resilience as well as the required computational complexity.

Raptor Code Ensembles

The finite length analysis of Raptor code ensembles under optimal erasure decoding, comprising the derivation of proper bounds on the residual erasure probability or not to mention the exact erasure probabilities, still remains a challenging open problem. Nevertheless, a practical method has been proposed to reliably estimate the erasure correction performance of these compound code ensembles. The so-called kernel weight profile thereby acts as an approximation of the erasure weight profile. Using the latter quantity, which is an important characteristic of an LT code ensemble and unfortunately can only be determined by extensive measurements, it is possible to judge whether an LT code ensemble is suitable for precoding. Additionally, by combining the erasure weight profile with the erasure correction performance of a precode, the compound erasure resilience can be calculated. With the kernel weight profile at least a very good approximation hereto could be obtained, which for practical values acts like an upper bound on the compound residual erasure probability.

Unequally Loss-Resilient LT Code Ensembles

For the transmission of data with unequally important parts, like hierarchically encoded multimedia content, ensembles are required which allow for an unequal erasure protection (UEP). Two methods from the literature have been reviewed and generalised to the non-binary domain.

The first considered approach for constructing UEP LT code ensembles is the weighting method. As the original weighting method possesses only discrete and irregularly spaced effective weights, not all protection levels are attainable. By introducing a variation in the form of biased sampling of input nodes, the accuracy of the weighting method has been improved and the achievability of any desired protection level has been rendered possible. A set of four bounds has been derived for a fixed but arbitrary importance class. Moreover, a heuristic has been presented to facilitate the design of UEP sparse random LT code ensembles, which are constructed by the weighting method with biased sampling, by means of the bounds of equivalent EEP sparse random LT code ensembles.

The second method to construct UEP LT code ensembles is the expanding window approach, for which the originally binary bounds on symbol level have been generalised to higher order Galois fields. Additionally, corresponding bounds on word level have been derived.

Appendix A

Further Proofs

For the sake of completeness the detailed proofs of the bounds on the residual erasure probability after optimal decoding for LT code ensembles with unequal erasure protection (UEP) in Chapter 7 shall be provided in this appendix. First, the bounds for the weighted UEP construction method with biased sampling are proved, which are stated in Corollaries 7.1 to 7.4. The proofs of the bounds for the expanding window UEP approach, as given by Corollaries 7.5 to 7.8, follow thereafter.

Weighted Unequal Erasure Protection with Biased Sampling of Input Nodes

Proof of Corollary 7.1. The probability $P_{\tau}^{[\mathfrak{L}]}\big(\cancel{W}\big)$ is equal to the probability that the kernel of $\mathbf{G}_{\mathrm{R}}$ is non-trivial w.r.t. importance class τ (referred to as τ-non-trivial in the following), i.e.

$$P_{\tau}^{[\mathfrak{L}]}\big(\cancel{W}\big) = \Pr\{\exists \mathbf{x} \in \ker(\mathbf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\},\, w_{\tau} = \|\mathbf{x}_{\tau}\|_{\mathrm{H}} \geq 1\}. \tag{A.1}$$

Similar to the proof of Theorem 3.14, the above expression is upper bounded by the expected cardinality of the τ-non-trivial kernel. And as before, just one of the $q-1$ non-trivial multiples is counted which results in a division by $q-1$:

$$\begin{aligned} P_{\tau}^{[\mathfrak{L}]}\big(\cancel{W}\big) &\leq \overline{P}_{\tau}^{[\mathfrak{L}]}\big(\cancel{W}\big) && \text{(A.2)}\\ &= \frac{1}{q-1} \cdot \mathrm{E}\{|\ker(\mathbf{G}_{\mathrm{R}}) \setminus \{\mathbf{0}\}|,\, w_{\tau} \geq 1\} && \text{(A.3)}\\ &= \frac{1}{q-1} \cdot \sum_{\substack{\mathbf{x} \in \mathbb{F}_q^k,\\ w_{\tau} \geq 1}} \Pr\{\mathbf{G}_{\mathrm{R}}\mathbf{x} = \mathbf{0}\}. && \text{(A.4)} \end{aligned}$$

The $k\gamma_{\mathrm{R}}$ rows of $\mathbf{G}_{\mathrm{R}}$ can be viewed as the outcomes of independent trials of a random variable $\mathbf{r} \in \mathbb{F}_q^k$.

$$\overline{P}_{\tau}^{[\mathfrak{L}]}\big(\cancel{W}\big) = \frac{1}{q-1} \cdot \sum_{\substack{\mathbf{x} \in \mathbb{F}_q^k,\\ w_{\tau} \geq 1}} \Pr\{\mathbf{r}\mathbf{x} = \mathbf{0}\}^{k\gamma_{\mathrm{R}}} \tag{A.5}$$

Now, the probability $\Pr\left\{\mathbf{r}^{\mathsf{T}}\mathbf{x}=0\right\}$ is determined, conditioned on

$$(\|\mathbf{r}_1\|_{\mathrm{H}},\ldots,\|\mathbf{r}_T\|_{\mathrm{H}})^{\mathsf{T}}=(d_1,\ldots,d_T)^{\mathsf{T}}=\mathbf{d}^{\mathsf{T}} \tag{A.6}$$

and

$$(\|\mathbf{x}_1\|_{\mathrm{H}},\ldots,\|\mathbf{x}_T\|_{\mathrm{H}})^{\mathsf{T}}=(w_1,\ldots,w_T)^{\mathsf{T}}=\mathbf{w}^{\mathsf{T}}. \tag{A.7}$$

There are

$$(q-1)^{w-1}\left(\prod_{i=1}^{T}\binom{k_i}{w_i}\right) \tag{A.8}$$

choices of $\mathbf{x}$ of overall weight $w>0$, with class specific weights $(w_1,\ldots,w_T)^{\mathsf{T}}$ and excluding the $q-1$ non-trivial multiples of $\mathbf{x}$. The class specific row weights $(\|\mathbf{r}_1\|_{\mathrm{H}},\ldots,\|\mathbf{r}_T\|_{\mathrm{H}})^{\mathsf{T}}=(d_1,\ldots,d_T)^{\mathsf{T}}$ occur with probability $\Omega_{d_1,\ldots,d_T}$ as given by the multivariate row weight distribution (7.6) which is based on the multivariate Wallenius' non-central hypergeometric distribution (7.3). So the expression for $P_{\tau}^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right)$ can be reformulated to

$$\begin{aligned}\overline{P}_{\tau}^{[\mathfrak{L}]}\left(\cancel{\mathcal{W}}\right)=&\sum_{\substack{w=w_1+\ldots+w_T=1\\ w_\tau\geq 1}}^{k}(q-1)^{w-1}\left(\prod_{i=1}^{T}\binom{k_i}{w_i}\right)\\ \cdot&\left[\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}}\Omega_{\mathbf{d}}\Pr\left\{\mathbf{r}^{\mathsf{T}}\mathbf{x}=0\left|\begin{pmatrix}\|\mathbf{r}_1\|_{\mathrm{H}}\\ \vdots\\ \|\mathbf{r}_T\|_{\mathrm{H}}\end{pmatrix}=\mathbf{d},\begin{pmatrix}\|\mathbf{x}_1\|_{\mathrm{H}}\\ \vdots\\ \|\mathbf{x}_T\|_{\mathrm{H}}\end{pmatrix}=\mathbf{w}\right.\right\}\right]^{k\gamma_{\mathrm{R}}}.\end{aligned} \tag{A.9}$$

Let $\mathbf{v}=(\mathsf{v}_1,\mathsf{v}_2,\ldots,\mathsf{v}_k)^{\mathsf{T}}$ with $\mathsf{v}_j=\mathsf{r}_jx_j$, where v_j, r_j and x_j are the j^{th} elements of the vectors $\mathbf{v}$, $\mathbf{r}$ and $\mathbf{x}$, respectively. Further, let $\mathbf{v}$ be also split up into class specific vectors $\mathbf{v}=(\mathbf{v}_1^{\mathsf{T}},\ldots,\mathbf{v}_T^{\mathsf{T}})^{\mathsf{T}}$ with $\mathbf{v}_\tau^{\mathsf{T}}=(\mathsf{v}_{\tau,1},\ldots,\mathsf{v}_{\tau,k_\tau})^{\mathsf{T}}$, then

$$\begin{aligned}&\Pr\left\{\mathbf{r}^{\mathsf{T}}\mathbf{x}=0\left|\begin{pmatrix}\|\mathbf{r}_1\|_{\mathrm{H}}\\ \vdots\\ \|\mathbf{r}_T\|_{\mathrm{H}}\end{pmatrix}=\mathbf{d},\begin{pmatrix}\|\mathbf{x}_1\|_{\mathrm{H}}\\ \vdots\\ \|\mathbf{x}_T\|_{\mathrm{H}}\end{pmatrix}=\mathbf{w}\right.\right\}\\ &=\sum_{l=l_1+\ldots+l_T=0}^{d}\Pr\left\{\begin{pmatrix}\|\mathbf{v}_1\|_{\mathrm{H}}\\ \vdots\\ \|\mathbf{v}_T\|_{\mathrm{H}}\end{pmatrix}=\begin{pmatrix}l_1\\ \vdots\\ l_T\end{pmatrix}\left|\begin{pmatrix}\|\mathbf{r}_1\|_{\mathrm{H}}\\ \vdots\\ \|\mathbf{r}_T\|_{\mathrm{H}}\end{pmatrix}=\mathbf{d},\begin{pmatrix}\|\mathbf{x}_1\|_{\mathrm{H}}\\ \vdots\\ \|\mathbf{x}_T\|_{\mathrm{H}}\end{pmatrix}=\mathbf{w}\right.\right\}\\ &\cdot\prod_{i=1}^{T}\Pr\left\{\sum_{j=1}^{k_i}\mathsf{v}_{i,j}=0\left|\,\|\mathbf{v}_i\|_{\mathrm{H}}=l_i\right.\right\}.\end{aligned} \tag{A.10}$$

The probability of occurrence of exactly l_i non-zero elements in $\boldsymbol{\mathsf{v}}_i$ is

$$\Pr\left\{\left(\begin{array}{c}\|\boldsymbol{\mathsf{v}}_1\|_{\mathrm{H}}\\ \vdots\\ \|\boldsymbol{\mathsf{v}}_T\|_{\mathrm{H}}\end{array}\right)=\left(\begin{array}{c}l_1\\ \vdots\\ l_T\end{array}\right)\middle|\left(\begin{array}{c}\|\boldsymbol{\mathsf{r}}_1\|_{\mathrm{H}}\\ \vdots\\ \|\boldsymbol{\mathsf{r}}_T\|_{\mathrm{H}}\end{array}\right)=\mathbf{d},\left(\begin{array}{c}\|\mathbf{x}_1\|_{\mathrm{H}}\\ \vdots\\ \|\mathbf{x}_T\|_{\mathrm{H}}\end{array}\right)=\mathbf{w}\right\}=\frac{\binom{w_i}{l_i}\binom{k_i-w_i}{d_i-l_i}}{\binom{k_i}{d_i}}, \tag{A.11}$$

while the last term in (A.10) is given by (cf. (3.34))

$$\Pr\left\{\sum_{j=1}^{k_i}\mathsf{v}_{i,j}=0\,\middle|\,\|\boldsymbol{\mathsf{v}}_i\|_{\mathrm{H}}=l_i\right\}=\frac{1}{q}\Big[1-(1-q)^{1-l_i}\Big]. \tag{A.12}$$

Finally, inserting (A.11) and (A.12) into (A.10) and the resulting expression into (A.9) yields (7.9) which concludes the assertion. □

Proof of Corollary 7.2. The probability $P_\tau^{[\mathfrak{L}]}\big(\not{S}\big)$ is equal to the probability that the j^{th} input symbol in $\mathbf{x}_\tau$ cannot be determined by ML decoding for an arbitrary $j \in \{1, 2, \ldots, k_\tau\}$

$$\begin{aligned} P_\tau^{[\mathfrak{L}]}\big(\not{S}\big) &= \Pr\Big\{\exists\mathbf{x}\in\mathbb{F}_q^k\setminus\{\mathbf{0}\},\, x_{\tau,j}=a\ :\ \boldsymbol{\mathsf{G}}_{\mathrm{R}}\mathbf{x}=\mathbf{0}\Big\} && \text{(A.13)}\\ &= \Pr\{\exists\mathbf{x}\in\ker(\boldsymbol{\mathsf{G}}_{\mathrm{R}})\setminus\{\mathbf{0}\},\, x_{\tau,j}=a\}. && \text{(A.14)}\end{aligned}$$

with arbitrary but fixed $a \in \mathbb{F}_q \setminus \{0\}$. The right-hand side of (A.13) is the probability of the j^{th} column of matrix $\boldsymbol{\mathsf{G}}_{\mathrm{R}}$ being linearly dependent on a non-empty set of columns. The expressions (A.14) and (A.1) differ merely in the number of choices of $\mathbf{x}$, which in the current case is

$$(q-1)^{w-1}\left(\prod_{i=1}^{T}\binom{k_i-\delta_{i,\tau}}{w_i-\delta_{i,\tau}}\right), \tag{A.15}$$

where $\delta_{i,\tau}$ is the Kronecker delta function, which equals one if $i = \tau$ and zero otherwise. The remainder of this proof follows the same line of thought as the proof of Corollary 7.1. □

Proof of Corollary 7.3. The partial information word in importance class τ cannot be reconstructed if at least one input node from importance class (IC) τ cannot be recovered. A lower bound on the residual word erasure probability $P_\tau^{[\mathfrak{L}]}\big(\not{W}\big)$ is therefore given by the probability that there exist input nodes in importance class τ that are not connected to any of the $k\gamma_\mathrm{R}$ independent output nodes

$$\underline{P}_\tau^{[\mathfrak{L}]}\big(\cancel{W}\big) = \Pr\{\exists \text{ IN in IC } \tau \text{ not connected to any ON}\} \tag{A.16}$$

$$= \sum_{j=1}^{k_\tau} \Pr\{\text{exactly } \mathsf{j} = j \text{ IN in IC } \tau \text{ not connected to any ON}\} \tag{A.17}$$

$$= \sum_{i=1}^{k_\tau} (-1)^{i+1} \binom{k_\tau}{i} \Pr\{\mathsf{i} = i \text{ particular IN in IC } \tau \text{ not connected to any ON}\}, \tag{A.18}$$

where the last line results from the arguments in the proof of Theorem 3.18. Rephrasing Lemma 3.16 for the multivariate case and for importance class τ yields

$$\Pr\{\mathsf{i} = i \text{ particular IN in IC } \tau \text{ not connected to any ON}\}$$

$$= \left(\sum_{d=d_1+\ldots+d_T=1}^{d_\mathrm{max}} \Omega_\mathbf{d} \frac{\binom{k_\tau - i}{d_\tau}}{\binom{k_\tau}{d_\tau}} \right)^{k\gamma_\mathrm{R}}. \tag{A.19}$$

Finally, inserting (A.19) into (A.18) results in (7.11). □

Proof of Corollary 7.4. Analogously to the binary EEP case [Sho06], a lower bound on $P_\tau^{[\mathfrak{L}]}\big(\not{S}\big)$ is given by the probability that an input node in class τ is not connected to any of the $k\gamma_\mathrm{R}$ output nodes. □

Expanding Window Unequal Erasure Protection

Proof of Corollary 7.5. The probability $P_\tau^{[\mathfrak{L}]}\big(\cancel{W}\big)$ is equal to the probability that the kernel of $\boldsymbol{\mathsf{G}}_\mathrm{R}$ is τ-non-trivial, i.e. it is non-trivial w.r.t. importance class τ:

$$P_\tau^{[\mathfrak{L}]}\big(\cancel{W}\big) = \Pr\{\exists \mathbf{x} \in \ker(\boldsymbol{\mathsf{G}}_\mathrm{R}) \setminus \{\mathbf{0}\},\, w_\tau = \|\mathbf{x}_\tau\|_\mathrm{H} \geq 1\}. \tag{A.20}$$

Similar to the proofs of Theorem 3.14 and Corollary 7.1, the above expression is upper bounded by the expected cardinality of the τ-non-trivial kernel. And as before, just one of the $q-1$ non-trivial multiples is counted which results in a division by $q-1$:

$$P_\tau^{[\mathfrak{L}]}\big(\cancel{W}\big) \leq \overline{P}_\tau^{[\mathfrak{L}]}\big(\cancel{W}\big) \tag{A.21}$$

$$= \frac{1}{q-1} \cdot \mathrm{E}\{|\ker(\boldsymbol{\mathsf{G}}_\mathrm{R}) \setminus \{\mathbf{0}\}|,\, w_\tau \geq 1\} \tag{A.22}$$

$$= \frac{1}{q-1} \cdot \sum_{\substack{\mathbf{x} \in \mathbb{F}_q^k, \\ w_\tau \geq 1}} \Pr\{\boldsymbol{\mathsf{G}}_\mathrm{R}\mathbf{x} = \mathbf{0}\}. \tag{A.23}$$

The $k\gamma_{\mathrm{R}}$ rows of $\mathbf{G}_{\mathrm{R}}$ can be viewed as the outcomes of independent trials of a random variable $\mathbf{r} \in \mathbb{F}_q^k$.

$$\overline{P}_{\tau}^{[\mathfrak{L}]}\big(\cancel{W}\big) = \frac{1}{q-1} \cdot \sum_{\substack{\mathbf{x} \in \mathbb{F}_q^k, \\ w_\tau \geq 1}} \Pr\{\mathbf{r}\mathbf{x} = \mathbf{0}\}^{k\gamma_{\mathrm{R}}} \tag{A.24}$$

Now, the probability $\Pr\big\{\mathbf{r}^{\mathsf{T}}\mathbf{x} = 0\big\}$ is determined, conditioned on

$$\|\mathbf{r}\|_{\mathrm{H}} = d \tag{A.25}$$

and

$$\big(\|\mathbf{x}_1\|_{\mathrm{H}},\, \|\mathbf{x}_2\|_{\mathrm{H}}, \ldots,\, \|\mathbf{x}_T\|_{\mathrm{H}}\big)^{\mathsf{T}} = \left(w_1 - w_0,\, w_2 - w_1, \ldots,\, w_T - w_{T-1}\right)^{\mathsf{T}}. \tag{A.26}$$

There are

$$(q-1)^{w_T - 1}\left(\prod_{i=1}^{T}\binom{k_i}{w_i - w_{i-1}}\right) \tag{A.27}$$

choices of $\mathbf{x}$ with the window specific weight $w_\tau \geq 1$ and excluding the $q-1$ non-trivial multiples of $\mathbf{x}$. Note that in the expanding window approach the class specific weights are given by $w_\tau - w_{\tau-1}$. The row weights $\|\mathbf{r}\|_{\mathrm{H}} = d$ are sampled according to the window specific row weight distribution $\Omega_i(\xi)$ and a window is selected according to the window selection distribution $\mathcal{W}(\xi)$. So the expression for $P_{\tau}^{[\mathfrak{L}]}\big(\cancel{W}\big)$ can be reformulated to

$$\begin{aligned}\overline{P}_{\tau}^{[\mathfrak{L}]}\big(\cancel{W}\big) = \sum_{w_T=1}^{\kappa_T} \cdots \sum_{w_\tau=1}^{\kappa_\tau} \sum_{w_{\tau-1}=0}^{\kappa_{\tau-1}} \cdots \sum_{w_1=0}^{\kappa_1} (q-1)^{w_T-1}\left(\prod_{i=1}^{T}\binom{k_i}{w_i - w_{i-1}}\right) \\ \cdot \left[\sum_{j=1}^{T}\mathcal{W}_j \sum_{d=1}^{\kappa_j}\Omega_{j,d}\Pr\left\{\mathbf{r}^{\mathsf{T}}\mathbf{x} = 0 \,\middle|\, \|\mathbf{r}\|_{\mathrm{H}} = d, \begin{pmatrix}\|\mathbf{x}_1\|_{\mathrm{H}} \\ \vdots \\ \|\mathbf{x}_T\|_{\mathrm{H}}\end{pmatrix} = \begin{pmatrix} w_1 - w_0 \\ \vdots \\ w_T - w_{T-1}\end{pmatrix}\right\}\right]^{k\gamma_{\mathrm{R}}}.\end{aligned} \tag{A.28}$$

Let $\mathsf{v} = (\mathsf{v}_1, \mathsf{v}_2, \ldots, \mathsf{v}_k)^{\mathsf{T}}$ with $\mathsf{v}_j = \mathsf{r}_j x_j$, where v_j, r_j and x_j are the j^{th} elements of the vectors v, $\mathbf{r}$ and $\mathbf{x}$, respectively. Further, let $\mathsf{v}_j = (\mathsf{v}_1, \mathsf{v}_2, \ldots, \mathsf{v}_{\kappa_j})^{\mathsf{T}}$ be the part of $\mathbf{v}$ assigned to window j of size κ_j, then

$$\Pr\left\{\mathbf{r}^{\mathsf{T}}\mathbf{x}=0 \,\middle|\, \|\mathbf{r}\|_{\mathrm{H}}=d, \begin{pmatrix} \|\mathbf{x}_1\|_{\mathrm{H}} \\ \vdots \\ \|\mathbf{x}_T\|_{\mathrm{H}} \end{pmatrix} = \begin{pmatrix} w_1-w_0 \\ \vdots \\ w_T-w_{T-1} \end{pmatrix}\right\}$$

$$=\sum_{l=0}^{d}\Pr\left\{\|\mathbf{v}_j\|_{\mathrm{H}}=l \,\middle|\, \|\mathbf{r}\|_{\mathrm{H}}=d, \begin{pmatrix} \|\mathbf{x}_1\|_{\mathrm{H}} \\ \vdots \\ \|\mathbf{x}_T\|_{\mathrm{H}} \end{pmatrix} = \begin{pmatrix} w_1-w_0 \\ \vdots \\ w_T-w_{T-1} \end{pmatrix}\right\}$$

$$\cdot\Pr\left\{\sum_{i=1}^{\kappa_j}\mathsf{v}_i=0 \,\middle|\, \|\mathbf{v}_j\|_{\mathrm{H}}=l\right\}. \tag{A.29}$$

The probability of occurrence of exactly l_j non-zero elements in $\mathbf{v}_j$ is

$$\Pr\left\{\|\mathbf{v}_j\|_{\mathrm{H}}=l \,\middle|\, \|\mathbf{r}\|_{\mathrm{H}}=d, \begin{pmatrix} \|\mathbf{x}_1\|_{\mathrm{H}} \\ \vdots \\ \|\mathbf{x}_T\|_{\mathrm{H}} \end{pmatrix} = \begin{pmatrix} w_1-w_0 \\ \vdots \\ w_T-w_{T-1} \end{pmatrix}\right\} = \frac{\binom{w_j}{l}\binom{\kappa_j-w_j}{d-l}}{\binom{\kappa_j}{d}}, \tag{A.30}$$

while the last term in (A.29) is given by (cf. (3.34))

$$\Pr\left\{\sum_{i=1}^{\kappa_j}\mathsf{v}_i=0 \,\middle|\, \|\mathbf{v}_j\|_{\mathrm{H}}=l\right\} = \frac{1}{q}\left[1-(1-q)^{1-l}\right]. \tag{A.31}$$

Finally, inserting (A.30) and (A.31) into (A.29) and the resulting expression into (A.28) yields (7.13) which concludes the assertion. □

Proof of Corollary 7.6. The probability $P_{\tau}^{[\mathfrak{L}]}\big(\not{\$}\big)$ is equal to the probability that the j^{th} input symbol in $\mathbf{x}_\tau$ cannot be determined by ML decoding for an arbitrary $j\in\{1,\,2,\ldots,\,k_\tau\}$

$$P_{\tau}^{[\mathfrak{L}]}\big(\not{\$}\big) = \Pr\big\{\exists\mathbf{x}\in\mathbb{F}_q^k\setminus\{\mathbf{0}\},\,x_{\tau,j}=a\,:\,\mathbf{G}_{\mathrm{R}}\mathbf{x}=\mathbf{0}\big\} \tag{A.32}$$

$$= \Pr\{\exists\mathbf{x}\in\ker(\mathbf{G}_{\mathrm{R}})\setminus\{\mathbf{0}\},\,x_{\tau,j}=a\}. \tag{A.33}$$

with arbitrary but fixed $a\in\mathbb{F}_q\setminus\{0\}$. The right-hand side of (A.32) is the probability of the j^{th} column of matrix $\mathbf{G}_{\mathrm{R}}$ being linearly dependent on a non-empty set of columns. The expressions (A.33) and (A.20) differ only in the number of choices of $\mathbf{x}$, which in this case is

$$(q-1)^{w_T-1}\left(\prod_{i=1}^{T}\binom{k_i-\delta_{i,\tau}}{w_i-w_{i-1}-\delta_{i,\tau}}\right), \tag{A.34}$$

The remainder of this proof follows the same line of thought as the proof of Corollary 7.5. □

Proof of Corollary 7.7. The partial information word in importance class (IC) τ cannot be reconstructed if at least one input node from IC τ cannot be recovered. A lower bound on the residual word erasure probability $P_\tau^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\big)$ is therefore given by the probability that there exist input nodes in IC τ that are not connected to any of the $k\gamma_{\mathrm{R}}$ independent output nodes

$$\underline{P}_\tau^{[\mathfrak{L}]}\big(\cancel{\mathcal{W}}\big) = \Pr\{\exists \text{ IN in IC } \tau \text{ not connected to any ON}\} \tag{A.35}$$

$$= \sum_{j=1}^{k_\tau} \Pr\{\text{exactly } \mathsf{j} = j \text{ IN in IC } \tau \text{ not connected to any ON}\} \tag{A.36}$$

$$= \sum_{i=1}^{k_\tau} (-1)^{i+1} \binom{k_\tau}{i} \Pr\{\mathsf{i} = i \text{ particular IN in IC } \tau \text{ not connected to any ON}\}, \tag{A.37}$$

where the last line results from the arguments in the proof of Theorem 3.18. Rephrasing Lemma 3.16 for the expanding window approach and for IC τ yields

$$\Pr\{\mathsf{i} = i \text{ particular IN in IC } \tau \text{ not connected to any ON}\}$$
$$= \left(\sum_{j=\tau}^{T} \mathcal{W}_j \sum_{d=1}^{\kappa_\tau} \Omega_{j,d} \sum_{d_\tau=0}^{k_\tau} \frac{\binom{k_\tau - i}{d_\tau}\binom{\kappa_j - k_\tau}{d - d_\tau}}{\binom{\kappa_j}{d}} \right)^{k\gamma_{\mathrm{R}}}, \tag{A.38}$$

where the first sum counts only those windows j which contain IC τ, the second sum denotes the probability of sampling a row weight d, given that IC τ is comprised in window j, and the third sum equals the probability that i particular input nodes in IC τ are not connected to any output node, given that window j comprises IC τ and given a row weight of d. Finally, inserting (A.38) into (A.37) results in (7.15). □

Proof of Corollary 7.8. Analogously to the binary EEP case [Sho06], a lower bound on $P_\tau^{[\mathfrak{L}]}\big(\cancel{\mathcal{S}}\big)$ is given by the probability that an input node in class τ is not connected to any of the $k\gamma_{\mathrm{R}}$ output nodes. □

Bibliography

[3GP13] 3GPP Technical Specification 26.346 V12.0.0. “Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS); Protocols and Codecs”, December 2013.

[AL96] N. Alon and M. Luby. “A Linear Time Erasure-Resilient Code with Nearly Optimal Recovery”. *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1732 –1736, November 1996.

[AS08] R. C. Alamino and D. Saad. “Typical Kernel Size and Number of Sparse Random Matrices over Galois Fields: A Statistical Physics Approach”. *Physical Review E*, vol. 77, pp. 061123, June 2008.

[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.

[BGL13] F. L. Blasco, G. Garrammone, and G. Liva. “Parallel Concatenation of Non-Binary Linear Random Fountain Codes with Maximum Distance Separable Codes”. *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4067–4075, 2013.

[BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima. “Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes (1)”. *IEEE International Conference on Communications (ICC)*, pp. 1064–1070, Geneva, Switzerland, May 1993.

[BKW97] J. Blömer, R. M. Karp, and E. Welzl. “The Rank of Sparse Random Matrices over Finite Fields”. *Random Structures and Algorithms*, vol. 10, no. 4, pp. 407–419, July 1997.

[BL11] F. L. Blasco and G. Liva. “On the Concatenation of Non-Binary Random Linear Fountain Codes with Maximum Distance Separable Codes”. *IEEE International Conference on Communications (ICC)*, pp. 1–5, Kyoto, Japan, June 2011.

[BLMR98] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. “A Digital Fountain Approach to Reliable Distribution of Bulk Data”. *ACM*

SIGCOMM Computer Communication Review, vol. 28, pp. 56–67, October 1998.

[BM04] D. Burshtein and G. Miller. “Efficient Maximum-Likelihood Decoding of LDPC Codes over the Binary Erasure Channel”. *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2837–2844, 2004.

[Bre14] T. Breddermann. *On Rate-Compatible Insertion Convolutional Turbo Codes and HARQ for Mobile Communications.* PhD thesis, IND, RWTH Aachen, January 2014.

[Bus52] K. A. Bush. “Orthogonal Arrays of Index Unity”. *Annals of Mathematical Statistics*, vol. 23, pp. 426–434, 1952.

[Cal96] N. J. Calkin. “Dependent Sets of Constant Weight Vectors in GF(q)”. *Random Structures and Algorithms*, vol. 9, no. 1-2, pp. 49–53, August/September 1996.

[Cal97] N. J. Calkin. “Dependent Sets of Constant Weight Binary Vectors”. *Combinatorics, Probability and Computing*, vol. 6, pp. 263–271, August 1997.

[Che76] J. Chesson. “A Non-Central Multivariate Hypergeometric Distribution Arising from Biased Sampling with Application to Selective Predation”. *Journal of Applied Probability*, vol. 13, no. 4, pp. 795–797, 1976.

[Coo00a] C. Cooper. “On the Distribution of Rank of a Random Matrix over a Finite Field”. *Random Structures and Algorithms*, vol. 17, no. 3-4, pp. 197–212, October-December 2000.

[Coo00b] C. Cooper. “On the Rank of Random Matrices”. *Random Structures and Algorithms*, vol. 16, no. 2, pp. 209–232, March 2000.

[Cop93] D. Coppersmith. “Solving Linear Equations over GF(2): Block Lanczos Algorithm”. *Linear Algebra and its Applications*, vol. 192, no. 0, pp. 33 – 60, 1993.

[Cop94] D. Coppersmith. “Solving Homogeneous Linear Equations over GF(2) via Block Wiedemann Algorithm”. *Mathematics of Computation*, vol. 62, no. 205, pp. pp. 333–350, 1994.

[Di04] C. Di. *Asymptotic and Finite-Length Analysis of Low-Density Parity-Check Codes.* PhD thesis, École Polytechnique Fédérale de Lausanne, 2004.

[Dor83] B. Dorsch. “Successive Check Digits Rather than Information Repetition”. *IEEE International Conference on Communications (ICC)*, pp. 323–327, Boston, MA, USA, June 1983.

[DPT$^+$02] C. Di, D. Proietti, I. Telatar, T. Richardson, and R. Urbanke. “Finite-Length Analysis of Low-Density Parity-Check Codes on the Binary Erasure Channel”. *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1570–1579, 2002.

[DRU06] C. Di, T. J. Richardson, and R. L. Urbanke. “Weight Distribution of Low-Density Parity-Check Codes”. *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 4839–4855, 2006.

[Eli55] P. Elias. “Coding for Noisy Channels”. *IRE Convention Record, Part IV*, pp. 37–46, 1955.

[ER63] P. Erdős and A. Rényi. “On Random Matrices”. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences (Series A)*, vol. 8, pp. 455–461, 1963.

[ETS09a] ETSI Technical Specification 102.034 V1.4.1. “Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks”, August 2009.

[ETS09b] ETSI Technical Specification 102.472 V1.3.1. “Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Content Delivery Protocols”, June 2009.

[Fog08a] A. Fog. “Calculation Methods for Wallenius’ Noncentral Hypergeometric Distribution”. *Communications in Statistics - Simulation and Computation*, vol. 37, no. 2, pp. 258–273, 2008.

[Fog08b] A. Fog. “Sampling Methods for Wallenius’ and Fisher’s Noncentral Hypergeometric Distributions”. *Communications in Statistics - Simulation and Computation*, vol. 37, no. 2, pp. 241–257, 2008.

[Fog11] A. Fog. “BiasedUrn: Biased Urn Model Distributions”. `http://cran.r-project.org/web/packages/BiasedUrn/index.html`, August 2011. Version 1.04.

[Gal63] R. G. Gallager. *Low-Density Parity-Check Codes.* M.I.T. Press, Cambridge, MA, USA, 1963.

[GMS08] K. M. Greenan, E. L. Miller, and T. J. E. Schwarz. “Optimizing Galois Field Arithmetic for Diverse Processor Architectures and Applications”. *IEEE International Symposium on Modeling, Analysis* *and Simulation of Computers and Telecommunication Systems (MASCOTS)*, pp. 1–10, Baltimore, MD, USA, September 2008.

[Gou72] H. W. Gould. *Combinatorial Identities: A Standardized Set of Tables Listing 500 Binomial Coefficient Summation*. Morgantown, West Virginia, 1972.

[Gou10] H. W. Gould. “Combinatorial Identities: Table I: Intermediate Techniques for Summing Finite Series”. `http://www.math.wvu.edu/~gould/Vol.4.PDF`, May 2010.

[GRT08] B. Geiser, S. Ragot, and H. Taddei. “Embedded Speech Coding: From G.711 to G.729.1”. R. Martin, U. Heute, and C. Antweiler, editors, *Advances in Digital Speech Transmission*, chapter 8, p. 201–247. John Wiley & Sons, Ltd., January 2008.

[Ham50] R. W. Hamming. “Error Detecting and Error Correcting Codes”. *The Bell System Technical Journal*, vol. 29, pp. 147–160, April 1950.

[Hub98] J. Huber. “Galoisfelder, das mathematische Werkzeug der algebraischen Kanalcodierung”. *Tagungsband zum Kurs 2 "Quellen- und Kanalcodierung für digitale Kommunikationssysteme" der Ferienakademie im Sarntal 1998*, pp. 119–182. Lehrstuhl für Informationsübertragung der Friedrich-Alexander-Universität Erlangen-Nürnberg, Lehrstuhl für Nachrichtechnik der Technischen Universität München, 1998.

[IT06] ITU-T. “ITU-T Rec. G.729.1: G.729-based embedded variable bit-rate coder: An 8-32 kbit/s scalable wideband coder bitstream interoperable with G.729”, 2006.

[IT13] ITU-T. “ITU-T Rec. H.264: Advanced Video Coding for Generic Audiovisual Services”, April 2013.

[KLF01] Y. Kou, S. Lin, and M. Fossorier. “Low-Density Parity-Check Codes Based on Finite Geometries: A Rediscovery and New Results”. *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2711–2736, 2001.

[Kol94] V. F. Kolchin. “Random Graphs and Systems of Linear Equations in Finite Fields”. *Random Structures and Algorithms*, vol. 5, no. 1, pp. 135–146, 1994.

[Kol99] V. F. Kolchin. *Random Graphs*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1999.

[Kol09] V. F. Kolchin. *Random Graphs*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2009.

[KPD$^+$08] K. Kasai, C. Poulliat, D. Declercq, T. Shibuya, and K. Sakaniwa. "Weight Distribution of Non-Binary LDPC Codes". *International Symposium on Information Theory and its Applications (ISITA)*, Auckland, New Zealand, December 2008.

[KPDS11] K. Kasai, C. Poulliat, D. Declercq, and K. Sakaniwa. "Weight Distributions of Non-binary LDPC Codes". *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 94-A, no. 4, pp. 1106–1115, 2011.

[Lan93] G. Landsberg. "Ueber eine Anzahlbestimmung und eine damit zusammenhängende Reihe". *Journal für die reine und angewandte Mathematik*, vol. 111, pp. 87–88, 1893.

[Lev05] A. A. Levitskaya. "Systems of Random Equations over Finite Algebraic Structures". *Cybernetics and Systems Analysis*, vol. 41, pp. 67–93, 2005.

[LH06] I. Land and J. Huber. *Information Combining.* Foundations and Trends in Communications and Information Theory. Now Publishers, November 2006.

[LHHH05] I. Land, S. Huettinger, P. Hoeher, and J. Huber. "Bounds on Information Combining". *IEEE Transactions on Information Theory*, vol. 51, no. 2, pp. 612–619, Feb 2005.

[LMPC09] G. Liva, B. Matuz, E. Paolini, and M. Chiani. "Pivoting Algorithms for Maximum Likelihood Decoding of LDPC Codes over Erasure Channels". *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–6, 2009.

[LMS$^+$97] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann. "Practical Loss-Resilient Codes". *29th ACM Symposium on Theory of Computing*, El Paso, Texas, USA, 1997.

[LMS98] M. Luby, M. Mitzenmacher, and A. Shokrollahi. "Analysis of Random Processes via And-Or Tree Evaluation". *9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[LNC97] R. Lidl, H. Niederreiter, and P. M. Cohn. *Finite Fields.* Cambridge University Press, 1997.

[LO91] B. LaMacchia and A. Odlyzko. "Solving Large Sparse Linear Systems over Finite Fields". A. Menezes and S. Vanstone, editors, *Advances in Cryptology - CRYPTO '90*, vol. 537 of *Lecture Notes in Computer Science*, pp. 109–133. Springer, 1991.

[LPC10] G. Liva, E. Paolini, and M. Chiani. “Performance versus Overhead for Fountain Codes over $\mathbb{F}_q$”. *IEEE Communications Letters*, vol. 14, no. 2, pp. 178–180, 2010.

[LPC13] G. Liva, E. Paolini, and M. Chiani. “Bounds on the Error Probability of Block Codes over the q-Ary Erasure Channel”. *IEEE Transactions on Communications*, vol. 61, no. 6, pp. 2156–2165, June 2013.

[LSW$^+$11] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder. “RaptorQ Forward Error Correction Scheme for Object Delivery”. Internet Engineering Task Force RFC 6330, August 2011.

[LSWS07] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer. “Raptor Forward Error Correction Scheme for Object Delivery”. Internet Engineering Task Force RFC 5053, October 2007.

[Lub01] M. Luby. “Information Additive Code Generator and Decoder for Communication Systems”. U.S. Patent 6,307,487, October 2001.

[Lub02a] M. Luby. “LT Codes”. *IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 271–280, Vancouver, BC, Canada, November 2002.

[Lub02b] M. Luby. “Information Additive Code Generator and Decoder for Communication Systems”. U.S. Patent 6,373,406, April 2002.

[Lup11] R. Lupoaie. “Flexible Error Protection for Scalable Video Streams Using Digital Fountain Codes”. Master thesis, Institute of Communications Systems and Data Processing (IND), RWTH Aachen University, September 2011.

[Mac97] D. MacKay. “Good Error-Correcting Codes Based on Very Sparse Matrices”. *IEEE International Symposium on Information Theory (ISIT)*, p. 113, Ulm, Germany, June–July 1997.

[Mac99] D. MacKay. “Good Error-Correcting Codes Based on Very Sparse Matrices”. *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[Mac05] D. MacKay. “Fountain Codes”. *IEE Proceedings Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.

[Man74] D. M. Mandelbaum. “An Adaptive-Feedback Coding Scheme Using Incremental Redundancy”. *IEEE Transactions on Information Theory*, vol. 20, no. 3, pp. 388–389, May 1974.

[May02] P. Maymounkov. "Online Codes". Technical Report TR2002-833, Secure Computer Systems Group, New York University, November 2002.

[MN95] D. J. MacKay and R. M. Neal. "Good Codes based on Very Sparse Matrices". *Cryptography and Coding, 5th IMA Conference*, vol. 1025 of *Lecture Notes in Computer Science*, pp. 100–111, Cirencester, UK, December 1995. Springer.

[MN96] D. J. C. MacKay and R. M. Neal. "Near Shannon Limit Performance of Low Density Parity Check Codes". *IEE Electronics Letters*, vol. 32, no. 18, pp. 1645–1646, July 1996.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1977.

[Odl81] A. M. Odlyzko. "On the Ranks of Some (0, 1)-Matrices with Constant Row Sums". *Journal of the Australian Mathematical Society (Series A)*, vol. 31, no. 02, pp. 193–201, 1981.

[OSI94] OSI. *Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*, 1994. ISO/IEC 7498-1.

[PLMC12] E. Paolini, G. Liva, B. Matuz, and M. Chiani. "Maximum Likelihood Erasure Decoding of LDPC Codes: Pivoting Algorithms and Code Design". *IEEE Transactions on Communications*, vol. 60, no. 11, pp. 3209–3220, 2012.

[PNF04] H. Pishro-Nik and F. Fekri. "On Decoding of Low-Density Parity-Check Codes over the Binary Erasure Channel". *IEEE Transactions on Information Theory*, vol. 50, no. 3, pp. 439–454, 2004.

[PS92] C. Pomerance and J. W. Smith. "Reduction of Huge, Sparse Matrices over Finite Fields Via Created Catastrophes". *Experimental Mathematics*, vol. 1, pp. 89–94, 1992.

[QUA10] QUALCOMM Incorporated. "RaptorQ$^{\text{TM}}$ Technical Overview", 2010.

[RF04] N. Rahnavard and F. Fekri. "Unequal Error Protection Using Low-Density Parity-Check Codes". *IEEE International Symposium on Information Theory (ISIT)*, p. 449, Chicago, IL, USA, 2004.

[RPNF07] N. Rahnavard, H. Pishro-Nik, and F. Fekri. "Unequal Error Protection Using Partially Regular LDPC Codes". *IEEE Transactions on Communications*, vol. 55, no. 3, pp. 387–391, 2007.

[RS60] I. S. Reed and G. Solomon. "Polynomial Codes over Certain Finite Fields". *Journal of the Society for Industrial and Applied Mathematics (SIAM)*, vol. 8, no. 2, pp. 300–304, June 1960.

[RU01] T. Richardson and R. Urbanke. “Efficient Encoding of Low-Density Parity-Check Codes”. *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 638–656, 2001.

[RU08] T. Richardson and R. Urbanke. *Modern Coding Theory.* Cambridge University Press, March 2008.

[RVF07] N. Rahnavard, B. N. Vellambi, and F. Fekri. “Rateless Codes With Unequal Error Protection Property”. *IEEE Transactions on Information Theory*, vol. 53, no. 4, pp. 1521–1532, 2007.

[Sej09] D. Sejdinovic. *Topics in Fountain Coding.* PhD thesis, University of Bristol, 2009.

[SGV13] B. Schotsch, G. Garrammone, and P. Vary. “Analysis of LT Codes over Finite Fields under Optimal Erasure Decoding”. *IEEE Communications Letters*, vol. 17, no. 9, pp. 1826–1829, September 2013.

[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.

[Sho04] A. Shokrollahi. “Raptor Codes”. *IEEE International Symposium on Information Theory (ISIT)*, p. 36, Chicago, IL, USA, June 2004.

[Sho06] A. Shokrollahi. “Raptor Codes”. *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[SL05] A. Shokrollahi and M. Luby. “Systematic Encoding and Decoding of Chain Reaction Codes”. U.S. Patent 6,909,383, June 2005.

[SL11] A. Shokrollahi and M. Luby. *Raptor Codes.* Foundations and Trends in Communications and Information Theory. Now Publishers, 2011. http://www.qualcomm.com/media/documents/files/raptor-codes-foundations-and-trends-in-communications-and-information-theory.pdf.

[SL12] B. Schotsch and R. Lupoaie. “Finite Length LT Codes over $\mathbb{F}_q$ for Unequal Error Protection with Biased Sampling of Input Nodes”. *IEEE International Symposium on Information Theory (ISIT)*, Cambridge, MA, USA, July 2012.

[SLK05] A. Shokrollahi, S. Lassen, and R. Karp. “Systems and Processes for Decoding Chain Reaction Codes Through Inactivation”. U.S. Patent 6,856,263, February 2005.

[SLL06] A. Shokrollahi, S. Lassen, and M. Luby. "Multi-Stage Code Generator and Decoder for Communication Systems". U.S. Patent 7,068,729, June 2006.

[SLV11] B. Schotsch, R. Lupoaie, and P. Vary. "The Performance of Low-Density Random Linear Fountain Codes over Higher Order Galois Fields under Maximum Likelihood Decoding". *Annual Allerton Conference on Communication, Control and Computing*, pp. 1004–1011, Monticello, IL, USA, September 2011.

[SMW07] H. Schwarz, D. Marpe, and T. Wiegand. "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.

[Spi96] D. A. Spielman. "Linear-Time Encodable and Decodable Error-Correcting Codes". *IEEE Transactions on Information Theory*, vol. 42, pp. 388–397, 1996.

[SS96] M. Sipser and D. A. Spielman. "Expander Codes". *IEEE Transactions on Information Theory*, vol. 42, pp. 1710–1722, 1996.

[SSV11] B. Schotsch, H. Schepker, and P. Vary. "The Performance of Short Random Linear Fountain Codes under Maximum Likelihood Decoding". *IEEE International Conference on Communications (ICC)*, pp. 1–5, Kyoto, Japan, June 2011.

[Sta11] R. P. Stanley. *Enumerative Combinatorics*, vol. 1. Cambridge University Press, 2nd edition, 2011.

[Str06] G. Strang. *Linear Algebra and Its Applications.* Thomson Brooks/Cole Cengage learning, 2006.

[STV07] S. Shamai, I. E. Telatar, and S. Verdú. "Fountain Capacity". *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 4372–4376, 2007.

[SV12] B. Schotsch and P. Vary. "Design of Unequally Error Protecting Low-Density Random Linear Fountain Codes". *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sydney, Australia, September 2012.

[SVD$^+$09] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. Senk, and R. Piechocki. "Expanding Window Fountain Codes for Unequal Error Protection". *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2510–2516, 2009.

[Tan81] R. M. Tanner. "A Recursive Approach to Low Complexity Codes". *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[Ulr57] W. Ulrich. "Non-Binary Error Correcting Codes". *The Bell System Technical Journal*, vol. 36, pp. 1341–1388, November 1957.

[VS12] D. Vukobratovic and V. Stankovic. "Unequal Error Protection Random Linear Coding Strategies for Erasure Channels". *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1243–1252, 2012.

[VSS+07] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. Fagoonee, and Z. Xiong. "Scalable Data Multicast Using Expanding Window Fountain Codes". *Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, USA, September 2007.

[VSS+09] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. Stankovic, and Z. Xiong. "Scalable Video Multicast Using Expanding Window Fountain Codes". *IEEE Transactions on Multimedia*, vol. 11, no. 6, pp. 1094–1104, 2009.

[Wal63] K. T. Wallenius. *Biased Sampling: The Noncentral Hypergeometric Probability Distribution.* PhD thesis, Stanford University, 1963.

[Wie86] D. H. Wiedemann. "Solving Sparse Linear Equations over Finite Fields". *IEEE Transactions on Information Theory*, vol. 32, no. 1, pp. 54–62, 1986.

[Wil94] H. Wilf. *Generatingfunctionology.* Academic Press, 2nd edition, 1994.

[WLK95a] N. Wiberg, H.-A. Loeliger, and R. Kötter. "Codes and Iterative Decoding on General Graphs". *European Transactions on Telecommunications (ETT)*, vol. 6, no. 5, pp. 513–525, 1995.

[WLK95b] N. Wiberg, H.-A. Loeliger, and R. Kötter. "Codes and Iterative Decoding on General Graphs". *IEEE International Symposium on Information Theory (ISIT)*, p. 468, Whistler, Canada, September 1995.

[YLV+13] J. Yue, Z. Lin, B. Vucetic, G. Mao, and T. Aulin. "Performance Analysis of Distributed Raptor Codes in Wireless Sensor Networks". *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4357–4368, 2013.

[Yua12] X. Yuan. "Joint Source-Channel Decoding with Digital Fountain Codes". Master thesis, Institute of Communications Systems and Data Processing (IND), RWTH Aachen University, February 2012.

[ZLJR08] V. Zyablov, M. Loncar, R. Johannesson, and P. Rybin. “On the Erasure-Correcting Capabilities of Low-Complexity Decoded LDPC Codes with Constituent Hamming Codes”. *11th International Workshop on Algebraic and Combinatorial Coding Theory (ACCT08)*, pp. 338–347, Pamporovo, Bulgaria, June 2008.

[ZP74] V. V. Zyablov and M. S. Pinsker. “Decoding Complexity for Low-Density Parity-Check Codes Used for Transmission over a Channel with Erasures”. *Problems of Information Transmission*, vol. 10, no. 1, pp. 10–21, 1974.