Analysis and Design of Raptor Codes for Multicast Wireless Channels

Auguste Venkiah

N° d'ordre:

THESIS

Presented to obtain the degree of Doctor of Sciences of the University of Cergy-Pontoise
Speciality: Telecommunications

Analysis and Design of Raptor Codes for Multicast Wireless Channels

by

Auguste Venkiah

Équipe d'accueil :
Équipes Traitement des Images et du Signal (ETIS) – CNRS UMR 8051

The thesis jury is composed of:

Dr	P. Ciblat	
Pr.	P. Duhamel	
Pr	D. Declercq	Advisor
Pr	S. McLaughlin	
Dr	C. Poulliat	Co-advisor
Pr	J. Sayir	Reviewer
Pr.	A. Shokrollahi	Reviewer

Abstract

In this thesis, we investigate the optimization of Raptor codes for various channels of interest in practical wireless systems. First, we present an analytical asymptotic analysis of jointly decoded Raptor codes over a BIAWGN channel. Based on the analysis, we derive an optimization method for the design of efficient output degree distributions. We show that even though Raptor codes are not universal on other channels than the BEC, Raptor code optimized for a given channel capacity also perform well on a wide range of channel capacities when joint decoding is considered. Then, we propose a rate splitting strategy that is efficient for the design of finite length Raptor codes. We next investigate the extension of the analysis to the uncorrelated Rayleigh-fading channel with perfect channel state information (CSI) at the receiver, and optimize Raptor codes for quasi-static fading channels when CSI is available at the receiver but not at the transmitter. Finally, we show that in presence of imperfect CSI at the receiver, it is possible to improve the performance with no additional complexity, by using an appropriate metric for the computation of the LLR at the output of the channel.
In the second part of this thesis, we investigate the construction of efficient finite length LDPC codes. In particular, we present some improvements for the Progressive Edge-Growth algorithm that allow to construct minimal graphs. The proposed algorithm is used to construct protographs with large girth that perform well under iterative decoding. Moreover, we propose an efficient structured search procedure for the design of quasi-cyclic LDPC codes.

Contents

List of Figures

List of Tables

Abbreviations

ACK	acknowledge receipt packet
ARQ	Automatic Repeat reQuest
BEC	Binary Erasure Channel
BER	Bit Error Rate
BIAWGN	Binary Input Additive White Gaussian Noise (channel)
BMS	Binary Memoryless Symmetric (channel)
BSC	Binary Symmetric Channel
BP	Belief Propagation
BPSK	Binary Phase Shift Keying
cPEG	Circulant PEG (algorithm)
CSI	Channel State Information
CSIR	Channel State Information at the Receiver
DE	Density Evolution
DFT	Discrete Fourier Transform
DVB	Digital Video Broadcast
EXIT	EXtrinsic Information Transfer (chart)
FEC	Forward Error Correction
FER	Frame Error Rate
GA	Gaussian Approximation
HARQ	Hybrid Automatic Repeat reQuest
IC	Information Content
i.i.d.	independant and identically distributed
IRA	Irregular Repeat Accumulate (code)

IR-HARQ	Incremental-Redundancy Hybrid Automatic Repeat reQuest
LDPC	Low-Density Parity-Check (code)
LDR	Log-Density Ratio
LLR	Log-Likelihood Ratio
LT	Luby Transform (code)
MAP	Maximum A Posteriori
ML	Maximum Likelihood
MP	Message Passing
PBF	Packet Based Fountain (code)
pdf	probability density function
PEG	Progessive Edge-Growth (algorithm)
QAM	Quadrature Amplitude Modulation
QC-LDPC	Quasi-Cyclic LDPC
QPSK	Quadrature Phase Shift Keying
QS	Quasi-Static (channel)
RTT	Round Time Trip
SIRA	Structured IRA (code)
SPA	Sum-Product Algorithm
UDP	User Datagram Protocol

Introduction

Context and historical background

In his landmark paper [Sha48], Shannon introduced information as a mathematical framework for both the storage and the transmission of digital signals. He also introduced the concept of redundant channel coding as a method to achieve reliable communicaion on a noisy channel with known capacity. In particular, he proved that for sufficiently long codes arbitrarily reliable communication is possible at any coding rate below the capacity. But Shannon's proof relies on random coding, which is impractical because of its complexity. Since then, the challenge of channel coding has been to design practical codes that approach the channel capacity.

The algebraic approach to channel coding lead to the introduction of BCH and Reed Solomon codes. These codes are linear Maximum Distance Separable (MDS) codes, which means they achieve the largest minimum distance possible for any given codeword length and coding rate. However, they are not decodable with maximum likelihood (ML) algorithms and thus even though they have optimal error correction properties, their decoding complexity remains prohibitive applications with large block lengths.

The discovery of turbo codes in 1993 has been a major breakthrough for the improvement of coded transmission in real systems: the paper [BGT93] presents a practical coding scheme that approaches the channel capacity within 1 dB at a bit error rate (BER) of 10^{-6}. More importantly the authors have showed the potential of iterative decoding as a mean of approaching the channel capacity. Since then, there has been a tremendous amount of research on codes that are iteratively decodable and generally defined by graphs. These codes are decoded with iterative decoding algorithms that have linear complexity with respect to the codeword length, and provide optimal max-

imum *a posteriori* (MAP) decoding in the limit of infinite block length. Low-Density Parity-Check (LDPC) codes were originally presented by Gallager in his PhD thesis [Gal62] , but received little attention at that time. With the breakthrough of turbo codes, they were rediscovered and regained more attention [Mac99].

We briefly highlight some key achievements in the field of iterative decoding of LDPC codes. In [LMS97, LMS98], the authors analyzed the iterative decoder in the case of the binary erasure channel (BEC). The so called *concentration theorems* [LMSS01b, RU01] state that the performance of a code randomly chosen from a code ensemble converge to the expected performance of the code ensemble as the codeword length increases, which allows to study the performance of code ensembles rather than specific graph instances, allowing more flexibility for the design of potentially good codes.

The first class of codes that provably achieve the capacity of the BEC using iterative decoding are Tornado codes, presented in [LMSS01a]. Other distributions have been proposed [OS01, OS02] since then. In [LMSS98, LMSS01b], the authors showed that appropriately chosen irregular graphs can perform better than regular ones and proposed an optimization technique for the irregularity profile. The optimization of the irregularity profile has been since then extended for a variety of different channels, such as the BIAWGN channel [RSU01, CFRU01], the uncorrelated Rayleigh fading channel, [HSM01], the BIAWGN channel with erasures [HM03], OFDM channels [MDG04] or multi-user channels [ADU02, RD07].

Motivation of the work

At this point, one is tempted to think that – since capacity achieving/approaching codes have been proposed – the channel coding problem is solved. However, one important constraint used in the aforementionned works is that point-to-point communication is under consideration. Situations where a single source broadcasts data to multiple receivers do not fall into the point-to-point paradigm. These situations arise in video/audio streaming (*e.g.* Internet TV/radio), software update distribution (em e.g. anti-virus update, operating system patch) or in peer to peer (P2P) applications. For fixed rate schemes, where the coding rate must be chosen according to worst channel conditions, the receivers with good channels suffer from information delay and rate loss. Thus, there is an unavoidable trade-off between reliability (low coding rate) and efficiency (high coding rate).

As a solution to this trade-off, we investigate the use of rateless coding schemes. Whereas traditional block codes are characterized by their design rate and require puncturing (resp. prunning) to achieve higher (resp. lower) rates, a rateless code generates as many coded symbols as necessary, and dynamically adapts the rate. Any sufficiently large set of coded symbols allows to recover the data at the receiver. LT codes, introduced in [Lub02], were the first efficient rateless codes, proved to be asymp-

totically capacity achieving on the BEC but they suffer from an error floor phenomenon when the decoding complexity is bounded Raptor codes, introduced in [Sho06] as an extension of LT codes, are constructed from the concatenation of an LT code and a high rate outer block code called precode.

Raptor codes have been extensively studied for the BEC [Sho06], and are used at the application layer as an efficient protocol for multicast [V7.07]. However, it remains unclear whether Raptor codes can be used on noisy channels, as a solution to the multicast channel coding problem on the *physical* layer. In this thesis, we generalize existing optimization methods to design Raptor codes for various channels of interest in practical wireless systems.

The second objective of this thesis is related to small block length behaviors. Capacity achieving (or approaching) codes only approach the capacity in the limit of infinite block length. However, the design of efficient finite length LDPC or Raptor codes remains a challenge. At small to moderate block lengths, random constructions perform rather poorly, and the performance of a code is closely related to the girth of the graph, which is defined as the size of the shortest cycle of the underlying Tanner graph. A pseudorandom construction based on a progressive edge-growth (PEG) of the graph was proposed in [HEA05], which results in graphs that have higher girths compared to pre-existing random LDPC code construction techniques.

However, the PEG algorithm is based on a greedy approach, and suffers from many limitations. We present some improvements in the PEG algorithm which greatly improve the girth properties of the resulting graphs: given a graph size, they increase the girth g achievable by the algorithm, and when the girth cannot be increased, our modified algorithm minimizes the number of cycles of length g. Moreover, hardware implementation constraints require that the codes are highly structured such that *(i)* they have a simple description and can be stored with minimum memory requirements, *(ii)* they are efficiently encodable, *(iii)* the decoder can take advantage of the code structure for parallel decoding. Quasi-cyclic (QC) LDPC codes have these three remarkable properties have attracted much attention because they are hardware friendly properties. We present algorithms to construct QC-LDPC codes with good girth properties, and we investigate both a structured search procedure and a PEG based approach.

Overview of contributions

Chapter 2 addresses the analysis and optimization of Raptor codes.

- We characterize jointly decoded Raptor codes on the BIAWGN channel, uncorrelated Rayleigh-fading channels, and quasi-static fading channels.

- We derive an optimization method for the design of efficient Raptor codes and present a thorough analysis of the main design parameters.
- We propose a rate splitting strategy that is efficient for the design of finite length Raptor codes.
- We consider the design of LT codes as a special case of our joint decoding model, and give new results on LT codes. In particular, we characterize the error floor region of an LT code and propose an optimization method for the irregularity profile of the precode.
- We investigate the performance of Raptor codes over higher order modulations, and show that in presence of imperfect CSI at the receiver, it is possible to improve the performance with no additional complexity, by using an appropriate metric for the computation of the LLR at the output of the channel.

Chapter 3 deals with the construction of LDPC codes and includes the following original contributions:

- An improved version of the PEG algorithm is propoed, called RandPEG, that allows to construct minimal graphs of column-weight two, called *cages*. The RandPEG is extended by enforcing a circulant constraint, and used to lift protographs.
- A structured construction of QC graphs with column-weight two and girth 8 is presented. Remarkably, the structured search allows to near minimal graphs of girth 8.. Extended results on a structured construction of QC cycle codes of girth 12 are reported in appendix C.

1

Rateless coding for multicast communications

In this chapter, we introduce Raptor codes as a solution to the problem of channel coding for wireless multicast transmissions. We introduce the parametrization of Raptor codes and the associated decoding algorithms for the binary erasure channel (BEC) and the binary input additive white Gaussian noise (BIAWGN) channel.

1.1 Channel coding for multicast transmissions

In a communication system, errors can occur at any layer of the system: physical, link, network, transport or application layer. If an error can be detected and corrected by reliability techniques at the layer at which the error occurred, then the error will not be visible to the layers above. If not, then the error is usually presented to the layer above as a packet loss, which the system may then attempt to correct at the layer above. The allocation of the available resources for error correction between the different layers of the system is therefore an important question for optimal operation of the system.

For link layers with an underlying network structure such as Ethernet, packet discards may occur due to buffer overloads (*i.e.* congestion) or faults in link layer networking equipment. Even in networks with low average utilization there can be wide variations in the instantaneous traffic load, and it can be expected that "transient congestion" events will be observed even in well-engineered networks. Moreover, device failures, restarts and internal faults cause outages resulting in packet loss events at the network level.

Such losses can only be addressed by end-to-end reliability mechanisms, and tradi-

tional multicast protocols have the major drawback that they are based on best effort approaches, *i.e.* message delivery is not guaranteed.

Automatic Repeat reQuest (ARQ) protocols detect lost packets and send a request to the transmitter to repeat the packet. The most common example of this is Transport Control Protocol (TCP). In general, the receipt of a repeat packet requires at least one Round Trip Time (RTT) from the point at which the loss is detected and the repeat request sent. If losses are at a level where the loss of a repeat request or the repeated packet itself must be recovered, then somewhat more than two RTTs are required. This is because the loss of the repeat request or repeated packet can only be detected by expiration of a timer at the receiver. ARQ can also be applied to multicast, again with the same disadvantage as described above. ARQ for multicast also suffers from a number of scalability issues. At its simplest, clients simply request repeat attempts to send a given packet from the multicast server. This places a load on the server proportional to the number of clients multiplied by the number of losses. Repeat attempts may be sent to the whole multicast group – generating load across the whole network – or only by unicast to the clients that request them. However this last option increases the load on the server because it must send packets in proportion to the number of clients, not just process requests and send responses in proportion to the number of losses. Finally, when the number of users increases the server can be flooded with NACK packets, which is known as the "NACK implosion problem".

Forward Error Correction (FEC) at the application/transport layers generally refers to packet erasure correction techniques. In these techniques an amount of data is sent which is in total greater than the stream to be communicated, with the property that the stream can be reconstructed from any sufficiently large subset of the transmitted data. The stream is thus resilient to a certain amount of loss (at most the difference between the transmitted and the original data size). For fixed rate schemes, where the rate must be chosen according to worst channel conditions, the receivers with good channels suffer from information delay and rate loss. Thus, there is an unavoidable trade-off between reliability (low coding rate) and efficiency (high coding rate). Alternative strategies have been widely considered. Hybrid ARQ (HARQ) is an enhanced version of ARQ where data is precoded with an error correcting code. This has two opposing effects: *(i)* the code rate penalty decreases the throughput and *(ii)* the probability that the transmission succeeds is increased. Incremental Redundancy HARQ (IR-HARQ) adapts the error correcting rate to the channel conditions. The receiver asks the transmitter for additional parity bits when decoding is not successful. IR-HARQ scheme based on LDPC codes have been proposed (see e.g [SCV04, KHRM06]), but one disadvantage is that such schemes cannot operate below a design rate which is the rate of the mother code.

An alternative to IR-HARQ is the use of *rateless* coding. In a rateless setting, the transmitter produces a potentially limitless number of independent symbols and the receiver tries to decode the information block as it receives output symbols. When

the receiver has collected enough output symbols to recover the message, then it can disconnect, potentially saving transmission time and energy. Whereas, traditional block codes are characterized by their design rate and require puncturing to achieve higher rates [HM04, HKKM06], a rateless code achieves this very naturally by adapting the number of output symbols. Indeed, all the output symbols are independent in a rateless setting, and any sufficiently large set of output symbols allows to recover the data with high probability. The rateless scheme can be practically implemented with Fountain codes, which are a family of naturally rateless codes.

In [LF05], the authors show that a Fountain based protocol (FBP) performs better than TCP on very congestionned networks for the delivery of large files. When hosts use TCP but act in a selfish manner (*i.e.* they do not implement congestion control mechanisms and transmit at the highest possible rate), the total throughput of the network drops. Thus, the optimal strategy for each user (never reducing rates) is strongly suboptimal for the network, which is a problem known as Tragedy of the Commons in economics. With a simple model, the authors show that by using a FBP, a Nash equilibrium is reached when all hosts use the FBP (and behave in a selfish manner), but that this does *not* drive the network to collapse. The key idea is that with a FBP, all packets arriving to destination are useful, and there are no duplicate packets.

1.2 Rateless coding for multicast wireless transmissions

A very challenging aspect in modern mobile wireless applications is the very strong constraints on the energy consumption. To achieve low energy consumption, it is necessary to make an efficient use of the transmission channels, *i.e.* transmit at the highest possible rate given the channel conditions. One measure that is of particular interest for a communication system is its *throughput*, defined as the average number of data bits accepted at the receiver (or reliable transmitted) per time required for the transmission of a single bit. Reliable transmission is obtained by using FEC techniques that introduce redundancy in order to be robust to transmission errors, and therefore reduce the throughput. Typically, a mobile receiver wants to minimize the transmission time (*i.e.* the battery time), by receiving the information at the highest rate possible for each channel realization.

Many physical layer systems, include some form of reliability mechanism within their channel coding and modulation mechanisms. But in multicast scenarios each receiver sees a different channel, leading to different maximal coding rates. Whether channel coding schemes are best allocated to the physical layer mechanisms or divided between physical layer and higher layer mechanisms is an important engineering decision and there is often much to be gained from finding an appropriate balance between the two.

Raptor codes have proved to be a good transport/application layers FEC for communication on lossy channels or when no feedback channel is available. In this thesis, we

will investigate the potential of Raptor codes as a solution to FEC on the physical link.

1.3 Raptor codes

1.3.1 A historical perspective

Fountain codes were originally introduced [BLMR98, BLM02] to transmit efficiently over a binary erasure channel (BEC) with unknown erasure probability. LT codes are the first class of efficient fountain codes, introduced by Luby [Lub02]. For a given number K of input symbols, an LT code produces a potentially limitless number of distinct output symbols according to an output degree distribution, and the receiver recovers the input symbols from any set of $(1+\epsilon)K$ output symbols, where ϵ is the *reception overhead*, or *overhead*. High performance, *i.e.* small overhead, is achieved by designing good output degree distributions, and LT codes are proved to be asymptotically capacity achieving on the BEC [Lub02], [Sho06]: they can recover with arbitrarily high probability *all* the input symbols. However, in order to obtain arbitrary small decoding failure probability, the average degree of the output symbols has to grow at least logarithmically with K. Thus, performance is achieved at a decoding cost growing in $O(K\log(K))$. This complexity is too high to ensure linear encoding and decoding time which is a desired property for practical codes.

Raptor codes are a class of fountain codes introduced by Shokrollahi in [Sho06] as an extension of LT codes. A Raptor code is the concatenation of an LT code with an outer code, called precode, which is a high rate error correcting block code: The LT code must enable the receiver to recover a large enough proportion of input symbols, and the precode is in charge of recovering the fraction of input symbols unrecovered by the LT code. Thus, the error correction capability of the precode relaxes strong constraints on the fountain design, and allows to design output degree distributions of constant average degree *i.e.* with linear encoding and decoding time. In [May02], the author independently presented the idea of precoding to obtain linear decoding time codes.

Recently in [ES06], the authors characterized Raptor codes on general binary memoryless symmetric channels with information theoretic arguments. An important result is that *universal* Raptor codes do not exist on general binary memoryless symmetric (BMS) channels: if a Raptor code approaches the capacity of a BMS channel, then the fraction of output nodes of degree 2 converges to a quantity that depends on the channel parameter. In the particular case of the BEC, this quantity is in fact a constant, and therefore, Raptor codes are universal on the BEC: they can approach the capacity of the channel arbitrarily closely, independently of the channel parameter.

In [PNF06], the authors propose the construction of "generalized Raptor codes", *i.e.* allow the output degree distribution to vary as the output symbols are generated. This construction has the advantage that the resulting codes have the potential to

approach the capacity of a symmetric channel, in a rate compatible way. However, their arguments are essentially information theoretical, and they do not propose any optimization method for this new construction.

1.3.2 Parametrization of Raptor codes

We call *input symbols* the information symbols to be transmitted and *output symbols* the symbols produced by an LT code from the input symbols. A symbol can be reduced to a bit, or more generally, an element of a finite field or a packet. The input symbols are not transmitted over the channel, and the received output symbols are used at the receiver to recover the input symbols.

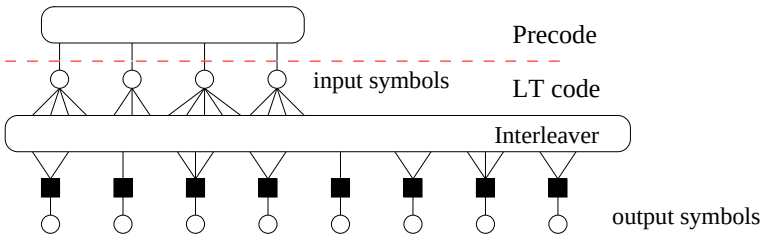


Figure 1.1 – Description of a Raptor code. Tanner graph of an LT code + precode. The black squares represent parity-check nodes and the circles are variable nodes associated with input symbols or output symbols.

An LT code is described by its *output degree distribution* [Lub02]: to generate an output symbol, a degree d is sampled from that distribution, independently from the past samples, and the output symbol is then formed as the sum of a uniformly randomly chosen subset of size d of the input symbols. This means that the d input symbols and the output symbol satisfy a parity-check equation. We say that the output symbol is of degree d.

A Raptor code can be represented by a Tanner graph, which is a bipartite representation of a system composed of variable nodes and parity-check nodes. The Tanner graph of a Raptor code is represented in Fig. 1.1. In a Tanner graph representation, the corresponding check node is of degree $d+1$: it is connected to d input symbols and 1 output symbol.

Let $\Omega_1, \Omega_2, \ldots, \Omega_{d_c}$ be the distribution weights on $1, 2, \ldots, d_c$ so that Ω_d denotes the probability of choosing the value d under this distribution. We denote the output (*node*) degree distribution using its generator polynomial:

$$\Omega(x) = \sum_{j=1}^{d_c} \Omega_j x^j$$

$\Omega(x)$ is associated with the following *edge* degree distribution in the Tanner graph:

$$\omega(x) = \frac{\Omega'(x)}{\Omega'(1)} = \sum_{j=1}^{d_c} \omega_j x^{j-1}$$

Because the input symbols are chosen uniformly at random, their node degree distribution is binomial, and can be well approximated by a Poisson distribution with parameter α [Sho06, ES06]. Thus, the input symbol *node* degree distribution is defined as:

$$I(x) = e^{\alpha(x-1)}$$

Then, the associated input symbol *edge* degree distribution is:

$$\iota(x) = \frac{I'(x)}{I'(1)} = e^{\alpha(x-1)}$$

Both distributions are of mean α. Technically, $I(x)$ and $\iota(x)$ cannot define degree distributions since they are power series and not polynomials. However, the power series can be truncated to obtain polynomials that are arbitrarily close to the exponential [ES06]:

$$I(x) = \sum_{i=1}^{d_v} I_i x^i$$

and

$$\iota(x) = \frac{I'(x)}{I'(1)} = \sum_{i=1}^{d_v} \iota_i x^{i-1}$$

A Raptor code is an LT code concatenated with an outer code called *precode*, which is a high rate error correcting block code. The input symbols of the LT code are formed by a codeword of the precode. Although it has been suggested for practical constructions [Sho06] to use a concatenation of Hamming codes and Low-Density Parity-Check (LDPC) code as precode, we consider the general definition where any high rate error correcting code can be considered as a precode, and focus on LDPC precoded Raptor codes. We do not consider concatenation with Hamming codes for the precode, because these constructions are specifically designed for the BEC case, and our results prove that with our approach, the use of Hamming codes is not necessary. Moreover, we restricted ourselves to *regular* LDPC precodes because for relatively high rates, regular codes are known to have good thresholds, close to the irregular thresholds.

1.3.3 A posteriori rate and transmission overhead

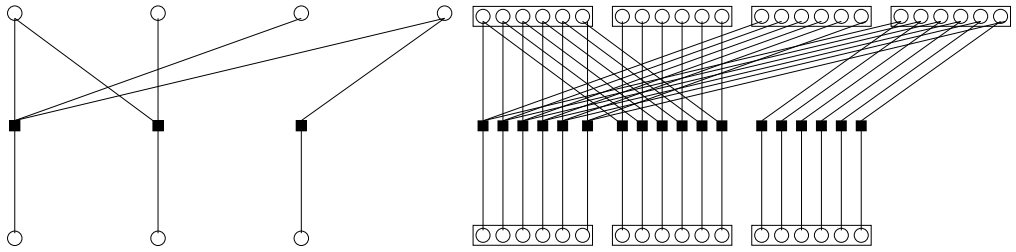
Although fountain codes are rateless, we can define an *a posteriori* rate R of a fountain code as follows:

$$R = \frac{\text{Nb input symbols}}{\text{Nb output symbols needed for successful decoding}} = \frac{\Omega'(1)}{\alpha}$$

For a channel capacity C, the rate R is associated to an overhead ϵ: $C = R(1+\epsilon)$ An overhead $\epsilon = 0$ means that the fountain code achieves the capacity, and an overhead of 0.1 means that it operates 10% away from the channel capacity. For the erasure channel, the overhead is defined with $C = 1$, because only the *received* output symbols appear in the decoding graph.

1.3.4 Packet based Fountain codes

In the general setting, the input/output symbols can be packets. Packet based fountains (PBF) have several advantages from a practical point of view.



(a) symbols are bits (b) symbols are packets

Figure 1.2 – This figure represents the same decoding graph: symbols in (a) are bits, whereas symbols in (b) are packets.

Using the same notations as defined in the 3GPP standard [V7.07], a symbol is defined as a packet of size T bits, and the packet size T is fixed by the transport layer. Under this setting, the Raptor code can encode a source of T KiB (1KiB = 8192 bits)[1]. For example, the setting $T = 512$ and $K = 8192$ enables to encode a source file of 512 KiB, and the setting $T = 1024$, $K = 8192$ enables to encode a source of 1 GiB. The symbol operations are defined packet-wise and the AND/OR operations used in the decoder recovery rule can be performed packet-wise. A Tanner graph of a PBF is represented in Fig. 1.2. For a PBF, the channel of interest is the Packet Erasure Channel (PEC), which is a realistic model for Internet communications.

Each recovered input symbol corresponds to a recovered packet. Similarly, all unrecovered symbols are unrecovered packets. This means that the performance of such Raptor codes is completely determined by the symbol error rate of the small decoding graph. In fact, decoding a PBF can be seen as decoding T parallel fountains, where all the fountain have the same decoding graph but with different data. A maximum value of $K_{MAX} = 8192$ is used for Raptor codes in the [V7.07] standard. This means that

[1]The term "kibibyte" (KiB) is recommended by the *NIST Reference on Constants, Units, and Uncertainty* to design the quantity 1024 bytes, because of the ambiguity with the kilobyte (KB) that designs the quantity 1000 bytes.

1	2	3	...	...	...	K
$K+1$	$K+2$	$K+3$	...	...	...	$2K$
$2K+1$	$2K+2$	$2K+3$	...	...	...	$3K$
...	...	...	...	...	...	...
$T.K+1$	$T.K+2$	$T.K+3$	...	...	...	$T.K$

Table 1.1 – The KT input bits are stored in a two dimensional $(K \times T)$ array in order to form K input packets of packet-size T

the decoding graph remains relatively small even with a large source size (determined by T), which means that decoding complexity remains small.

An alternative to PBF is to use a bit-wise fountain and form the packets after the fountain encoding. In that case, the decoding graph is much larger, hence the decoding complexity is larger. However, a larger fountain operates closer to its asymptotic regime and therefore exhibits better performance. Note that this discussion concerns iterative decoding only, and that with ML decoding excellent performance can be achieved even with small lengths.

Finally we point out that in a bit setting, it questionable to consider that bits arrive one at a time, and if a fountain code of source size $K = 1024$ bits produces output that arrive in a packets of, say, 256 bits, then the minimum overhead of practical interest is $256/1024 = 0.25$ %, which corresponds to 1 packet of overhead. However, in a PBF each output symbol is an output packet. Considering that packets arrive one at a time, it makes sense to consider in that case an overhead as small as 1 output symbol.

1.4 Decoding LT codes and Raptor codes

1.4.1 Decoding on the BEC

1.4.1.1 The Decoder Recovery Rule

At the receiver side, the output symbols are used to recover iteratively the input symbols. The procedure for decoding LT codes over a BEC, called the *decoder recovery rule* [Lub02], is an instance of the *peeling decoder* adapted to the decoding of LT codes. The peeling decoder, originally introduced for decoding LDPC codes over a BEC [LMS97], is an efficient implementation of the MP decoder that "peels off" one degree 1 check node per decoding iteration – hence the name – and has a decoding time that grows linearly with the number of edges in the decoding graph. As the variable nodes are recovered, the corresponding outgoing edges are removed from the decoding graph, and when the decoding procedure succeeds, there are no edges in the decoding graph.

When removing an edge from the graph at round (l), the degree of the associated output

symbols is decreased. We call *reduced degree* at round (l) the number of neighbors in the current setting. We call *ripple* the set of input symbols connected to at least one output symbol of reduced degree 1. An output symbol is *released* at round (l) if its reduced degree becomes 1 at round (l). When an output symbol is released, its neighbor is added to the ripple.

When an input symbol is in the ripple (if it is connected to an output symbol of reduced degree 1), then it can be recovered immediately since it is a copy of the output symbol.

The decoding procedure is described in Algorithm 1.

Algorithm 1: Decoder Recovery Rule

Ripple $\triangleq$ { input symbols connected to at least one output symbol of degree 1. }
while Ripple $\neq \emptyset$ **do**

1. Pick an input symbol in the ripple
2. Recover its value
3. Process the input symbol:
   - XOR the input symbol to all its remaining neighbors
   - remove the input symbol and all its outgoing edges in the graph.
4. Update the ripple

1.4.1.2 Maximum Likelihood decoding for the BEC

The decoder recovery rule is a very simple algorithm that enables to recover iteratively the input symbols in a computationally efficient way. Unfortunately, the decoder is not very efficient for small to moderate codeword lengths, and many input symbols can remain unrecovered if at some point of the decoding process the ripple vanishes.

However, the output symbols are defined as parity-check symbols of the input symbols, and as long as there are slightly more output symbols than input symbols, then the equation system that gives the input symbols as a linear function of the output symbols is almost surely full rank. This means that an ML decoder can recover the input symbols from the output symbols with Gaussian elimination.

Gaussian elimination is a computationally intensive task, and an efficient implementation of the ML decoding procedure as described in the 3GPP standard [V7.07] takes advantage of iterative decoding. When the ripple is empty, an input symbol of reduced degree 2 is marked as known and feeds the ripple, which enables the iterative process to go on. Gaussian elimination is used to recover the subset of input symbols marked as known, which form a system of linear equations, together with the parity-check

equations that must be satisfied by the precode. This means that the complexity of ML decoding reduces to the Gaussian elimination on a reduced subset of input symbols that cannot be recovered iteratively (which would be a stopping set for the LDPC case). This decoding algorithm is closely related to the Maxwell Decoder (M Decoder) described in [MMU04, RU07].

ML decoding allows to recover the input symbols with very little overhead (typically 1-2% overhead only), regardless of the codeword length since even with a small overhead, the underlying system is almost surely full rank.

1.4.2 Decoding on a noisy channel

In the case of a noisy channel, only message passing decoding can be considered, since there is no such algorithm as Gaussian elimination in the noisy case. The algorithm for decoding graph based codes, namely the Sum Product Algorithm, gives a maximum *a posteriori* (MAP) estimation of the transmitted codeword. MAP decoding of a graph based code boils down to computing marginals of multivariate functions, namely

$$\hat{x_i} = \arg\max_{x_i} p(x_i|y_1 \dots y_N)$$

where x_i denotes the i^{th} information bit and $y_1 \dots y_N$ denote the N channel observations. In that context, factor graphs [KFL01] provide a natural graphical description of the factorization of a global function into a product of local functions.

1.4.2.1 Factor graphs and the Sum-Product Algorithm (SPA)

Many problems can efficiently be described by a factor graph [FKLW97, KFL01]. For instance, they can be used to represent graphical codes such as LDPC codes, Raptor codes, convolutionnal or Turbo codes, but also estimation problems such as optimal filtering.

A factor graph is a bipartite graph that expresses how a global function of several variables factors into a product of local factors. The two types of vertices in the graph are called variable nodes and function nodes. The variable nodes can represent the value of a bit or a state in a Markov chain, whereas the function nodes represent the interactions between variable nodes.

The Sum Product Algorithm (SPA) is a general algorithm for computing marginals of the global function by distributed message passing (MP) in the corresponding factor graph [FKLW97, KFL01]. In the case of a cycle-free factor graph (when the graph is a tree), SPA is optimal in the sense that it produces the exact marginalization. When there are cycles in the factor graph, there is no guarantee that the SPA will converge at all. However, it has been observed that it actually does converge to densities that are relatively close to the actual marginal distributions.

Many well-known algorithms are in fact a particular instance of the SPA algorithm. When the factor graph under consideration is the graph of a state-space hidden Markov model, the corresponding instance of SPA used for MAP decoding of convolutionnal codes is called the BCJR algorithm [BCJR74]. In statistical inference, it is known as the "Forward-Backward" algorithm, and if all the probability distributions are Gaussian, it is also known as the "Kalman Filter".

With minor modification, by changing the "sum" operation to a "max" operation the algorithm becomes "max-product" algorithm, which, implemented in the negative log domain, is more widely known as the "min-sum algorithm". The specific instance of the min-sum algorithm on the factor graph of a state-space hidden Markov model is commonly known as the Viterbi [Vit37] algorithm. We refer the reader to [KFL01] for a detailed presentation of factor graphs and the SPA algorithm.

In this thesis, the function nodes under consideration are parity-check constraint nodes, the factor graph is called a Tanner graph.

1.4.2.2 Finite Alphabet Message Passing decoders

We consider the case where the messages on the graph are defined over a finite alphabet $\mathcal{A}$. The case $\mathcal{A} = \{-1, 1\}$ gives rise to numerous bit flipping algorithms, amongst which Gallager A and Gallager B [Gal62] are the most well-known representatives. These decoding algorithms have been widely studied for the decoding of LDPC codes (see e.g. [BRU04, AK05] and references therein) but cannot be used for decoding fountain codes, because the input symbols being unknown (erased) at the beginning of the decoding process, it is necessary to code for erasures in the decoder.

The Erasure Decoder, independently presented and analysed in [RU01] and [Mit98] can be used to decode Raptor codes over a channel with errors such as the BSC [MS06b]. When used to decode over a BEC, the Erasure Decoder is equivalent to the Decoder Recovery Rule decoder, presented in 1.4.1.1.

1.4.2.3 Belief Propagation

The belief propagation (BP) decoder is also a message passing decoder, where the messages propagated on the decoding graph are the log density ratios (LDRs) of the probability weights defined over $\mathbb{R}$. At each node on the graph, the messages are locally propagated according to Bayes rules.

1.5 Fountain based protocols

Unlike block codes (LDPC, turbo ...) the Tanner graph of a Raptor code is random. Therefore, the problem of constructing the graph at the decoder arises naturally because the graph description cannot be sent to the receiver as this description would be larger than the transmitted information itself ! A practical approach (used in 3GPP standard) is to use a seed synchronization technique: the seed of the random generator is sent within each packet header. Therefore, the receiver is able to use the same pseudorandom generator with the same seed to construct the pseudorandom graph. The random generator is standardized in [V7.07].

Fountain codes can be used in application layer or transport layer as an alternative to the File Transfer Protocol (FTP) for example. They can be used for point-to-point, point-to-multipoint (broadcast/multicast) or multipoint-to-multipoint communications (peer to peer (P2P) applications). The communication protocols need not be synchronous. New asynchronous multicast applications can be based on Fountain codes, such as asynchronous video on demand (VoD), or multiple rate congestion control.

In [MM03], the authors present a Fountain based protocol for peer to peer (P2P) communications, and show that Fountain codes are well adapted to multi-source download because of an *availability* property, defined as the ability to recover information from sources with partial knowledge of a file (which is the case when the source nodes with complete knowledge of the file leave the network).

MBMS is a point-to-multipoint IP-based service carried by the 3G air interface of W-CDMA (UMTS Terrestrial Radio Access Network or UTRAN) or the 2.5G air interface of EDGE/GPRS (GSM EDGE Radio Access Network or GERAN). Raptor codes are required as the sole mandatory application layer FEC protocol for multimedia broadcast in 3GPP MBMS technical specification [V7.07] and IP datacast over DVB-H [DVB06]. In the 3GPP specification, Raptor codes operate above the User Datagram Protocol (UDP) layer, and below the Real Time Protocol (RTP) layer. UDP does not guarantee reliability or ordering in the way that TCP does. Datagrams may arrive out of order, appear duplicated, or go missing without notice.

Fountain based coding schemes have been proposed in other different communication setting. For instance in [YCLX08], the authors show that Raptor codes provide a good solution for satellite communications, characterized by very long delays, high error rates, little bandwidth for a feedback channel. In [HST08], the authors investigate the potential of rateless codes for wireless reprogramming of a sensor network (Over-the-Air (OAP) programming), with large network size and high packet loss. They show that rateless coding is of great interest since it drastically reduces the need of packet rebroadcasting.

Currently, codes are only used on transport/application layers. A natural question that arises is whether or not Raptor codes are adapted for the use on a physical layer.

This is the problem that we tackle in this thesis. We will address the optimization of Raptor codes for various channels of interest for wireless communications, namely the BIAWGN channel, the uncorrelated Rayleigh fading channel, and the quasi-static Rayleigh fading channel.

2

Analysis and Design of Jointly Decoded Raptor Codes

In the previous chapter, we presented Raptor codes and showed how they can be used for efficient communication in a multicast scenario. In this chapter, we investigate the analysis and optimization of Raptor codes.

We assume that the two code components (LT code and precode) are decoded *jointly*, that is to say that the two component codes exchange extrinsic information at each decoding iteration. We present an asymptotic analysis of the joint decoder for the BIAWGN channel, the uncorrelated Rayleigh fading channel, and the quasi-static Rayleigh fading channel. The analysis of the joint decoder for the BEC is reported in appendix B. Based on the asymptotic analysis, we derive an optimization method. The optimization of a Raptor code consists in optimizing the rateless part of the Raptor code, namely its *output degree distribution* $\omega(x)$.

Finite length design is addressed with a rate splitting strategy. We show in particular that with almost no asymptotic loss, it is possible to design Raptor codes that perform very well at small to moderate lengths by using a relatively low rate precode.

Performance over higher order modulations is investigated for both the AWGN and the uncorrelated Rayleigh fading channels, when perfect channel state information (CSI) is available at the receiver. Then we show that in presence of imperfect CSI at the receiver, it is possible to improve the performance with no additional complexity, by using an appropriate metric for the computation of the LLR at the output of the channel.

2.1 Asymptotic analysis of graph based codes

The purpose of this section is to briefly present the tools that we shall use in the following sections for the asymptotic analysis of Raptor codes. We present the density evolution technique, and an efficient approximation of density evolution for the BIAWGN channel, namely the density evolution under Gaussian approximation. Information Content evolution is a monodimensional approach based on the Gaussian approximation, that is used for the analysis of the BP decoder.

2.1.1 Code ensembles

Graph-based codes such as Turbo codes or Low-Density Parity-Check (LDPC) codes have proved to operate very close to the channel capacity, when the codeword length tends to infinity. In that context, characterizing the performance of an explicit code is a dawning task. Consequently, a problem of great interest is to characterize the performance of a code *ensemble*. Often, this characterization can be performed in the asymptotic regime, *i.e.* as the codeword length increases to infinity.

A common characterization in channel coding is the minimum distance of a code, which enables to give the performance of a finite length code decoded with a maximum likelihood decoder. This characterization is no longer possible in the asymptotic regime. In contrast, the quantity of interest is the decoding threshold, defined as the highest noise level that can be handled by the code ensemble.

2.1.2 Density Evolution

The main tool for characterizing the performance of a given decoding algorithm defined on graph-based codes in the asymptotic regime is called density evolution (DE). DE consists in tracking the densities of the probability messages on the decoding graph through the decoding iterations, hence the name "density evolution".

The density evolution analysis mainly relies on the three following properties.

(a) Concentration theorem The "concentration around ensemble average" and the "convergence of ensemble average to the cycle-free case" are two main results that enable an asymptotic analysis. Roughly speaking, these results state that the performance of a code randomly sampled from a code ensemble converges to the expected performance of the code ensemble as the codeword length increases. The concentration theorems were stated in [LMSS01b], where transmission over the BEC and BSC is investigated, and the decoder is a hard decision message passing algorithm. The theorems was extended to the BIAWGN channel and BP decoding in [RU01].

(b) Conditional Independence An asymptotic analysis is sometimes referred to as an analysis under "treelike assumption", because in the asymptotic regime, the decoding graph of a node is locally a tree. More precisely, the local tree assumption states that the girth of the graph is large enough so that the local neighborhood of any variable node is a tree (there are no repeated nodes in the subgraph). When the local neighborhood of a node is a tree, then all messages passed through the graph are conditionally independent.

(c) Symmetry Let u be the random variable with probability density function (pdf) $f_U(u)$. A variable is *symmetric* if $f_U(u) = f_U(-u)e^u$. When the channel is output symmetric, or equivalently when the channel output pdf is symmetric, which is true for all binary-input *symmetric* memoryless channels, we say that the "symmetry condition" is satisfied. One important property is that the symmetry is preserved through the decoding iterations [RSU01, RU01], which means that when the pdf of the channel LLR is symmetric, then the marginal probability distributions computed with the SPA are symmetric.

It has been shown (see e.g. [RU01]) that when the symmetry condition is satisfied, the bit error probability of the code ensemble does not depend on the transmitted codeword. For all linear codes, the "all-zero word" is indeed a codeword and therefore one can assume without of loss of generality that the all-zero codeword is transmitted. This assumption greatly simplifies the density evolution analysis. Moreover, when the symmetry condition is satisfied, the convergence of the bit error probability to zero is equivalent to the convergence of the message density to the point mass at $+\infty$ (δ_∞).

2.1.3 Gaussian Approximation

Density Evolution has been derived for the general case of BMS channels including the BIAWGN channel in [RU01, RSU01], Unfortunately, DE is no longer a monodimensionnal approach in the case of the BIAWGN channel.[1] Even with an efficient implementation (Fast Density Evolution was proposed in [JR06]), DE remains computationally very challenging.

More importantly, DE and Fast DE do not provide any analytical characterization of the decoder, and thus they do not provide a mean to optimize degree distribution with linear programming. Rather, DE is used to compute exact thresholds of a code ensemble, or it can be associated to more sophisticated optimization techniques such as differential evolution or simulated annealing to optimize irregularity profiles.

A Gaussian approximation (GA) of DE is proposed in [CRU01]. It is shown to be a

[1] In the BEC case, DE is parametrized with the erase probability as a single parameter. For the BSC case and bit flipping algorithms (Gallager A,B), the message density is also monodimensionnal and is parametrized with the error probability as a single parameter.

faithful surrogate to full DE with far lower computational requirements, and provides a monodimensional characterization of BP decoder on the BIAWGN channel, without much sacrifice in accuracy. The method is based on approximating message densities as symmetric Gaussian densities. The messages on the decoding graph, namely the log density ratios (LDR) of the probability weights, are modeled by a random variable which is assumed to be Gaussian distributed with mean m. For a Gaussian distributed variable, enforcing the symmetry condition imposes that the variance equals two times the mean: $\sigma^2 = 2m$ [CRU01].

2.1.3.1 Mean evolution

With GA, the analysis of the BP decoder in [CRU01] becomes a monodimensionnal problem, and consists in tracking the evolution of LDRs mean through the decoding iterations, which can be given as a recursion. Let $v = \log\left(\frac{p(y|c=0)}{p(y|c=1)}\right)$ and $u = \log\left(\frac{p(y'|c'=0)}{p(y'|c'=1)}\right)$ denote the LDR messages at the output of a variable node and at the output of a check node respectively.

In this section, we assume that all the incoming messages at a node are i.i.d., and in particular, we do not distinguish the message from the channel.

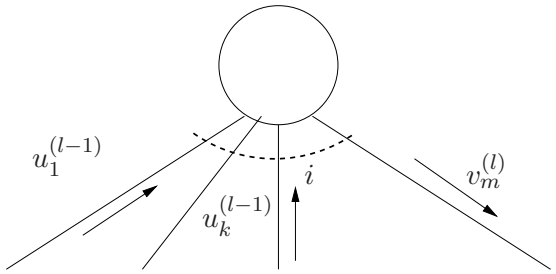


Figure 2.1 – Notations used for the message update rule at a variable node of degree i

The BP message update rule at a variable node of degree i is:

$$v_m^{(l)} = \sum_{k=1,k\neq m}^{i} u_k^{(l-1)}, \forall m = 1 \dots i \tag{2.1}$$

The BP message update rule at a parity-check constraint node of degree j is:

$$\tanh\frac{u_k^{(l)}}{2} = \prod_{m=1,m\neq k}^{j} \tanh\frac{v_m^{(l)}}{2}, \forall k = 1 \dots j \tag{2.2}$$

We denote the means of u and v by m_u and m_v respectively. Then taking the expectation of eq. (2.1) gives:

$$m_v^{(l)} = (i-1)m_u^{(l-1)} \tag{2.3}$$

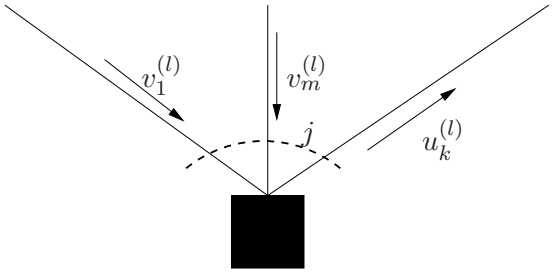


Figure 2.2 – Notations used for the message update rule at a parity-check constraint node of degree j

and eq. (2.2) becomes

$$E\Big[\tanh\frac{u^{(l)}}{2}\Big] = E\Big[\tanh\frac{v^{(l)}}{2}\Big]^{j-1} \tag{2.4}$$

Let $\phi(\cdot)$ be defined as follows:

$$\begin{aligned}\phi(x) &= 1 - E\Big[\tanh\frac{u}{2}\Big] \\ &= 1 - \frac{1}{\sqrt{4\pi x}}\int_R \tanh\frac{u}{2} e^{-\frac{(u-x)^2}{4x}} du\end{aligned} \tag{2.5}$$

then eq. (2.4) can be written:

$$1 - \phi(m_v^{(l)}) = \Big[1 - \phi(m_u^{(l-1)})\Big]^{j-1} \tag{2.6}$$

Since $\phi(\cdot)$ is continuous and monotonically decreasing on $[0;\infty]$, its inverse function $\phi^{-1}(\cdot)$ is well defined and it is possible to explicitly express $m_v^{(l)}$ as a function of $m_u^{(l-1)}$. Therefore, by combining (2.3) and (2.6), $m_v^{(l)}$ can be seen as a function of $m_v^{(l-1)}$, which is given as a recursive functional (see [CRU01] for details).

2.1.3.2 IC evolution

We call *information content* (IC), the mutual information between a random variable representing a transmitted bit and another one representing an LDR message on the decoding graph. IC evolution is another monodimensionnal analysis of the BP decoder that tracks the information content associated with the LDR messages on the decoding graph through the iterations, under symmetric and Gaussian assumption of the messages.

Similarly to [RGCV04], we define the binary-input symmetric-output capacity functional $\mathcal{J} : \mathcal{F}_{sym} \to [0;1]$, such that

$$\mathcal{J}(F) = 1 - \int_{\mathbb{R}} \log_2(1 + e^{-z}) dF(z) \tag{2.7}$$

Namely, $\mathcal{J}$ maps any symmetric distribution $\mathcal{F}$ into the capacity of the binary-input symmetric-output channel with transition probability $p_{Y|X}(y|0) = \mathcal{F}(y)$

For a BIAWGN channel, $\mathcal{F}$ is Gaussian symmetric distribution of mean m, and the IC associated to the LDR message is $x = J(m)$, where $J(\cdot)$ is defined by:

$$J(m) = 1 - \frac{1}{\sqrt{4\pi m}} \int_{\mathbb{R}} \log_2(1 + e^{-\nu}) \exp\left(- \frac{(\nu - m)^2}{4m} \right) d\nu \tag{2.8}$$

$J(\cdot)$ maps m to the capacity of a BIAWGN channel with output LLR distributed $\mathcal{N}(m, 2m)$. Equivalently, the capacity of a BIAWGN channel with noise variance σ^2 equals $J\left(\frac{2}{\sigma^2}\right)$.

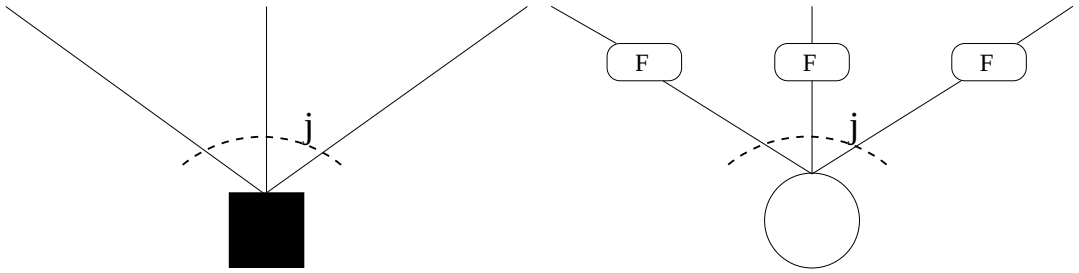
Recalling that u is a Gaussian and symmetric distributed random variable ($u \sim \mathcal{N}(m_u, 2m_u)$), let $x_u^{(l)}$ (resp. $x_v^{(l)}$) denote the IC associated to a message u (resp v). The IC evolution equations corresponding to eq. (2.1) are:

$$J^{-1}(x_v^{(l)}) = (i-1)J^{-1}(x_u^{(l-1)}) \tag{2.9}$$

Moreover, if $\mathcal{F}(\cdot)$ denotes the Fourier transform then it can be shown that

$$\mathcal{F}(U \otimes V) = \mathcal{F}(U).\mathcal{F}(V)$$

, which means that a check node acts in the Fourier domain as a variable node, which is illustrated in Fig. 2.1.3.2. Moreover, a simple computation shows that $x_{\mathrm{DFT}(u)} = 1 - x_u$,[2] then eq. (2.4) can be written as follows:



(a) Original flow (b) Equivalent flow

Figure 2.3 – Message flow through a parity-check constraint node.

$$J^{-1}(1 - x_v^{(l)}) = (j-1)J^{-1}(1 - x_u^{(l-1)}) \tag{2.10}$$

IC evolution is a concurrent tool of mean evolution under GA, that has been proved to be more accurate and robust for the optimization of LDPC/IRA codes [RGCV04]. In

[2] A similar statement on the BEC is called the "reciprocical channel" approximation

particular, the stability condition of an LDPC code derived from IC evolution model is exactly the same condition as the one given with true density evolution [Pou04], whereas the stability condition derived from mean evolution differs slightly [CRU01]. Moreover, thresholds of an LDPC code estimated with IC evolution are closer to the actual thresholds computed with density evolution than the thresholds computed with mean evolution [Dec03, Table 3.1].

2.2 Analysis of Raptor codes over the BIAWGN channel

The performance of LT codes and Raptor codes on a BIAWGN channel was first investigated in [PY04]. The authors show that output degree distributions designed for the BEC (from [Sho06]) perform reasonably well on a BIAWGN channel. In [ES06], the authors proposed an optimization procedure for designing good output degree distributions for the BIAWGN channel, based on mean evolution under a refined Gaussian approximation that requires Monte Carlo simulations [AK04]. In fact, the authors show that mean evolution under Gaussian approximation is not sufficient for the optimization of efficient output degree distributions.

In this thesis, we consider the joint decoding of the two code components, that has several advantages over the classical tandem decoding scheme, which we will show throughout the chapter. We derive an asymptotic analysis of the joint decoder of Raptor codes on a BIAWGN channel, presented from the fountain point of view. With joint decoding, we assume that extrinsic information is exchanged between the precode and the fountain at each decoding iteration. We will track the evolution of the IC of the messages that are related to the fountain part of the Tanner graph. Thus, in the same way that the optimization of LDPC codes boils down to the optimization of a left irregularity profile $\lambda(x)$, our objective is to optimize the output degree distribution of the fountain part of the Raptor code, namely $\omega(x)$, taking into account the contribution of the precode through its IC transfer function.

2.2.1 Joint decoder

The classical "tandem decoding" (TD) consists in decoding the LT code first and then using the soft extrinsic information about the input symbols as *a priori* information for the precode. With "joint decoding" (JD), one decoding iteration consists of alternating N_{lt} decoding iterations on the LT code, and N_p decoding iterations on the precode. Thus, both code components of the Raptor code, LT code and precode, provide extrinsic information to each other. In the sequel, we shall only consider the case where $N_{lt} = N_p = 1$, and where the precode is an LDPC code. When Joint Decoding is considered, the Raptor code can be described by a single Tanner graph with two kinds of parity-check nodes: check nodes of the precode, referred to as *static check nodes* and parity-check nodes of the LT code, later referred to as *dynamic check nodes* [Sho06].

Although joint decoding has already been proposed as a practical decoding scheme, all previous works rely on a tandem decoding scheme for the analysis and optimization of Raptor codes. As we shall see, joint decoding has several advantages over the classical tandem decoding scheme, and it is important to optimize the distributions for the joint decoding of the two code components by taking into account the contribution of the precode during the decoding process.

2.2.2 Information Content evolution

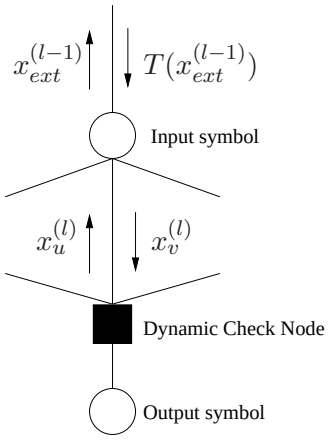


Figure 2.4 – Notations used for the asymptotic analysis of a Raptor code with IC evolution.

We denote $x_u^{(l)}$ (resp. $x_v^{(l)}$) the IC associated to messages on an edge connecting a dynamic check node to an input symbol (resp. an input symbol to a dynamic check node) at the l^{th} decoding iteration. Moreover, we denote by $x_{\text{ext}}^{(l-1)}$ the extrinsic information passed from the LT code to the precode, at the l^{th} decoding iteration, and $T(\cdot) : x \mapsto T(x)$ the IC transfer function of the precode. The extrinsic information passed by the precode to the LT code is then $T(x_{\text{ext}}^{(l)})$. The notations are summarized in Fig 2.4. When accounting for the transfer function of the precode, the IC update rules in the Tanner graph can be written as follows:

- Input symbol message update:

$$x_v^{(l)} = \sum_{i=1}^{d_v} \iota_i J\Big((i-1)J^{-1}(x_u^{(l-1)}) + J^{-1}\big(T(x_{ext}^{(l-1)})\big)\Big) \tag{2.11}$$

- Dynamic check node message update:

$$x_u^{(l)} = 1 - \sum_{j=1}^{d_c} \omega_j J\Big((j-1)J^{-1}(1-x_v^{(l)}) + f_0\Big) \tag{2.12}$$

  with: $f_0 \triangleq J^{-1}\big(1 - J\big(\frac{2}{\sigma^2}\big)\big) = J^{-1}\big(1-C\big)$ where C is the channel capacity.

- Precode extrinsic information update:

$$x_{ext}^{(l)} = \sum_i I_i J\big(iJ^{-1}(x_u^{(l)})\big) \tag{2.13}$$

Replacing (2.11) in (2.12) gives (2.15), the monodimensionnal recursive equation $x_u^{(l)} = F(x_u^{(l-1)}, \sigma^2, T(\cdot))$ that describes the evolution through one joint decoding iteration of the IC of the LDRs at the output of the dynamic check nodes (fountain part):

$$x_u^{(l)} = F(x_u^{(l-1)}, \sigma^2, T(.)) \tag{2.14}$$

$$= 1 - \sum_{j=1}^{d_v} \omega_j J\Bigg((j-1) J^{-1}\Big(1 - \sum_{i=1}^{d_c} \iota_i J\big((i-1)J^{-1}(x_u^{(l-1)}) + J^{-1}\big(T(x_{ext}^{(l-1)})\big)\big)\Big) + f_0 \Bigg) \tag{2.15}$$

Note that for a given distribution $\iota(x)$, this expression is linear with respect to the coefficients of $\omega(x)$, which is the distribution that we intend to optimize.

We point out here that (2.15) is general, since it reduces to the classical tandem decoding case by setting the extrinsic transfer function to $x \mapsto T(x) = 0 \quad \forall x \in [0;1]$, thus assuming that no information is exchanged between the precode and the fountain.

If the precode is an error correcting code that has a soft-input soft-output decoding algorithm, then its transfer function $x \mapsto T(x)$ can be estimated with Monte Carlo simulations. When the precode is an LDPC code, an analytical expression of the transfer function can be given. Let $\lambda(x)$ (resp. $\Lambda(x)$) denote the variable edge (resp. node) degree distribution and $\rho(x)$ the check edge degree distribution, then the IC transfer function [Bri01] is given by:

$$T(x) = \sum_{i=2}^{d_v} \Lambda_i J\Bigg(i J^{-1}\Big(1 - \sum_{j=2}^{d_c} \rho_j J\big((j-1)J^{-1}(1-x)\big)\Big)\Bigg) \tag{2.16}$$

Remark: we assume here a reinitialization of the decoder, which is a pessimistic assumption. Indeed, we assume that the initial messages from static check nodes to input symbol nodes are set to 0, which means that the values of the messages on the LDPC graph are not kept from one global iteration to another. Naturally, the messages in the LDPC decoder are not reinitialized from one iteration to another in a practical decoder. However, this pessimistic assumption is crucial to lead to a *linear* optimization problem, and proves sufficient for the design of efficient output degree distributions.

2.2.3 Fixed point characterization

In an IC evolution analysis, the convergence is guaranteed by $F(x, \sigma^2, T(.)) > x$. Convergence continues toward a fixed point of $x \mapsto F(x, \sigma^2, T(.))$. Unfortunately, there are no trivial solutions for the fixed point of (2.15). However, using a functional analysis, an upper bound on the fixed point can be given. Replacing $x_u^{(l-1)}$ by 1 (its maximal value) and using the fact that $T(1) = 1$ in (2.15), we obtain:

$$\lim_{x \to 1} F(x, \sigma^2, T(.)) = J\Big(\frac{2}{\sigma^2}\Big) \triangleq x_0 \tag{2.17}$$

which means that, because $x \mapsto F(x, \sigma^2, T(.))$ is an increasing function, the fixed point is necessarily less or equal than x_0, which is the capacity of a BIAWGN channel with parameter σ^2. Thus, the IC is upper bounded through the decoding iterations by x_0. This gives some insights on the asymptotic behavior at the decoding convergence point: the BP decoding of the LT part of a Raptor code is limited on a BIAWGN channel by the capacity of the channel.

This result is not really surprising and can be interpreted as follows: the output symbols have a constant contribution on each dynamic check node. As the iterative decoding process goes on, the IC of the messages at the output of the dynamic check nodes is limited by the reliability of the channel observations.

2.2.4 Starting condition

We now derive a simple condition for the beginning of the decoding process. If this condition is not met, then the decoding of a Raptor code is not possible.

Lemma 2.1 (Starting condition)**.** *The decoding process can begin iff* $F(0, \sigma^2, T(.)) > 0$ *and the following holds:*

$$F(0, \sigma^2, T(.)) > \varepsilon \quad \Longleftrightarrow \quad \omega_1 > \frac{\varepsilon}{J\big(\frac{2}{\sigma^2}\big)} \tag{2.18}$$

Proof. The decoding process can begin iff $x_u^{(1)} > \varepsilon$, for some arbitrarily small $\varepsilon > 0$. At the first iteration, $x_u^{(0)} = 0$, and (2.15) gives: $x_u^{(1)} = F(0, \sigma^2, T(.)) = \omega_1 J\big(\frac{2}{\sigma^2}\big)$ □

Therefore, one must have $\omega_1 > 0$ for the decoding process to begin, and ε appears to be a design parameter that will constrain the optimization problem, ensuring that $\omega_1 \neq 0$. In practice, the value of ε can be chosen arbitrarily small. Indeed, it has been proved [ES06] that for a sequence of capacity achieving distributions $\omega^{(n)}(x)$, $\lim_{n\to\infty} \omega_1^{(n)} = 0$.

Remark: as an illustration we point out that, for the "Ideal Soliton Distribution" introduced by Luby [Lub02], $\Omega_1 = 1/K$, which is the smallest proportion possible with K input symbols.

2.2.5 Lower bound on the edge proportion of degree 2 output symbols

In [ES06], an important bound on Ω_2, the proportion of output symbols of degree 2, is derived for sequences of capacity achieving distributions. Following steps of [ES06], we derive a similar bound for the proportion ω_2 of a capacity achieving distribution $\omega(x)$, specifically for the IC evolution method. The IC evolution equations represent

a dynamic system, where $x = 0$ is a fixed point for a capacity achieving distribution. This bound is derived by considering that it must *not* be an attractive fixed point, which is given by lemma 1.

Lemma 2.2. *For an output degree distribution that is to be capacity achieving, we have:*

$$F'(0, \sigma^2, T(.)) > 1 \tag{2.19}$$

Proof. For a sequence of capacity achieving output degree distributions indexed by n, $\lim_{n\to\infty} \omega_1^{(n)} = 0$ [ES06], which means that $\lim_{n\to\infty} F^{(n)}(0, \sigma^2, T(.)) = 0$. Then, the convergence condition $F(x, \sigma^2, T(.)) > x$ implies that $\lim_{x\to 0} F'(x, \sigma^2, T(.)) > 1$. □

According to the following proposition, this inequality gives a lower bound on the proportion of degree 2 output symbols:

Lemma 2.3. *When considering IC evolution, eq* (2.19) *gives the following necessary condition for a distribution* $\omega(x)$ *to be capacity achieving:*

$$\omega_2 > \frac{1}{\alpha e^{-f_0/4}} \tag{2.20}$$

Proof. see Appendix A □

This condition on the fraction of output nodes of degree 2 can be interpreted as a counterpart of the stability condition for LDPC codes [RSU01]. The stability condition for LDPC codes gives an upper bound on the fraction of variable nodes of degree 2, such that $x = 1$ is a stable fixed point for the decoder, that is to say that once the decoder is close enough to a successful convergence, then it will indeed converge to its fixed point. Here, the lower bound on the output nodes of degree 2 for a capacity achieving output degree distribution ensures that $x = 0$ is *not* an attractive fixed point of the decoder, that is to say that the decoder successfully starts.

2.2.6 Threshold of a Raptor code

In this section, we discuss the threshold behavior with the IC evolution model of jointly decoded Raptor codes, and compute numerically the exact thresholds of a Raptor code.

2.2.6.1 Threshold behavior of a Raptor code

Definition 2.1. *The a posteriori rate is the rate below which the decoding is successful. We define the threshold* ϵ^* *of a Raptor code as the asymptotic overhead corresponding to expectation of its a posteriori rate.*

We only consider the case such that the precode is a block error correcting code with a threshold behavior (an LDPC code for example). For tandem decoding, it is clear that the Raptor code has a threshold behavior: when LT code converges to its fixed point, it is sufficient that this fixed point is such that the extrinsic information passed to the precode is higher than the precodes threshold.

In the case of joint decoding, we adopt the same strategy, except that during the convergence of the extrinsic information passed from the fountain to the precode to its limiting value $x_u^{(\infty)}$, we assume belief propagation decoding on the whole Raptor code Tanner graph. The scheduling that we propose has then two steps: during the first step, the Raptor code is decoded under joint decoding, and the LT part of the Tanner graph converges to its fixed point. The convergence is guaranteed by eq. (2.15) under Gaussian approximation. During the second step, the precode is decoded alone, and the extrinsic information passed from the LT code is used as *a priori* information for the precode.

Since the precode is assumed to have a threshold, the joint decoding of a Raptor code with the proposed scheduling exhibits a threshold behavior.

2.2.6.2 Numerical estimation of thresholds

To estimate the threshold of a Raptor code, we use a numerical method that is an instance of Density Evolution (DE) described in Algorithm 2.

At each decoding iteration, the Tanner graph of the Raptor code is randomly permuted, and a new channel noise realization is considered. This instance of DE simulates an infinite graph by "breaking" the correlation between messages in the decoding graph, and ensuring that the channel is Gaussian distributed. When the decoding fails, we decrease the rate of the code. When the decoding is successful, we increase the rate of the code. This procedure is run several times and the consistency of the estimated threshold is reached by averaging.

Algorithm 2: Threshold estimation procedure

$\epsilon_{min} = 0$
$\epsilon_{max} = 1$
while $(\epsilon_{max} - \epsilon_{min} > 0.01)$
$\epsilon = (\epsilon_{min} + \epsilon_{min})/2$
`Generate a random graph of a Raptor code`
for $i = 1 : maxIter$

1. `Permute the precode interleaver`
2. `Permute the fountain code interleaver`
3. `New channel noise realization`
4. `BP Data Node message update`
5. `BP Check Node message update`

end for
if (decoding == successful)
$\epsilon_{max} = \epsilon$
else
$\epsilon_{min} = \epsilon$
end while
$\epsilon^* = (\epsilon_{max} + \epsilon_{min})/2$

2.3 Optimization of output degree distributions for joint decoding

In this section, we explicit the optimization problem for the design of good output degree distributions, and give some complementary results that we use for the choice of the design parameters. We will assume that the channel parameter σ^2 is given, that is to say that the output degree distribution is optimized for a given channel parameter. We will see that the optimization problem can be stated as a linear problem, and therefore can easily be solved with linear programming by algorithms such as the Simplex algorithm.

2.3.1 Optimization problem statement

The optimization of an output distribution consists in maximizing the a posteriori rate of the corresponding LT code, *i.e.* maximizing $\Omega'(1) = \sum_i \Omega_i i$, which is equivalent to minimizing $\sum_i \omega_i / i$. Moreover, according to the previous section, several constraints must be satisfied.

- [C_1] Since $\omega(x)$ is a probability distribution, its coefficients must sum up to 1. We call this the *proportion constraint* C_1.
- [C_2] To ensure the convergence of the iterative process, we must have $F(x, \sigma^2, T(.)) > x$. However, this inequality cannot hold for each and every value of x: the analysis in section 2.2.3 shows that the fixed point of $F(x, \sigma^2, T(\cdot))$ is smaller than $x_0 = J\left(\frac{2}{\sigma^2}\right)$. Therefore, we must fix a margin $\delta > 0$ away from x_0, and then by discretizing $[0; x_0 - \delta]$ and requiring inequality to hold on the discretization points, we obtain a set of inequalities that need to be satisfied: they define the *convergence constraint* C_2. The margin δ appears to be a design parameter, and its choice is discussed in section 2.3.3.
- [C_3] The starting condition (proposition 2.1) must also be satisfied and defines the *starting constraint* C_3.
- [C_4] The edge proportion of output symbols of degree 2 is lower bounded by proposition 2.3, defining the *stability constraint* C_4.

Finally, $x \mapsto T(x)$ is defined according to (2.16) for an LDPC code, or could be estimated with Monte Carlo simulations if another component code is used as a precode. The IC transfer function $T(\cdot)$ appears in the general IC evolution and therefore in constraint [C_2].

For a given value of α, and a given channel parameter σ^2, the cost function and the constraints are linear with respect to the unknown coefficients ω_i. Therefore, the optimization of an output degree distribution can be written as a linear optimization

problem that can be efficiently solved with linear programming. The optimization problem can be stated as follows:

$$\omega_{opt}(x) \quad = \quad \arg\min_{\omega(x)} \sum_j \frac{\omega_j}{j} \tag{2.21}$$

subject to the constraints:

[C_1] Proportion constraint: $\sum_i \omega_i = 1$

[C_2] Convergence constraint: $F(x, \sigma^2, T(.)) > x \quad \forall x \in [0; x_0 - \delta]$ for some $\delta > 0$

[C_3] Starting condition: $\omega_1 > \frac{\varepsilon}{J\left(2/\sigma^2\right)}$ for some $\varepsilon > 0$

[C_4] Flatness condition: $\omega_2 > \frac{1}{\alpha e^{-f_0/4}}$

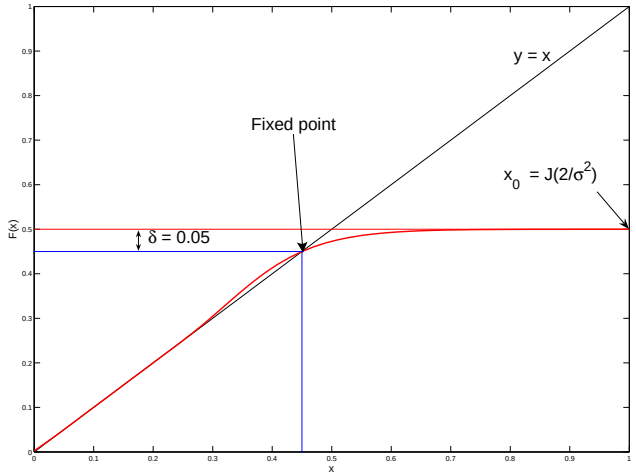


Figure 2.5 – $y = F(x, \sigma^2, T(\cdot))$: EXIT function of an output degree distribution optimized for a BIAWGN channel of capacity $C = 0.5$

2.3.2 Parameter α

The average degree of the input symbols α is the main design parameter. Indeed, for increasing values of the design parameter α, we optimized output degree distributions as explained in the previous section. The distributions are optimized for a BIAWGN channel of capacity $C = 0.5$ ($\sigma = 0.9786$), with a regular (3,60) precode of rate $R_p = 0.95$. Fig. 2.7 shows that there is a value for α that maximizes the corresponding rate

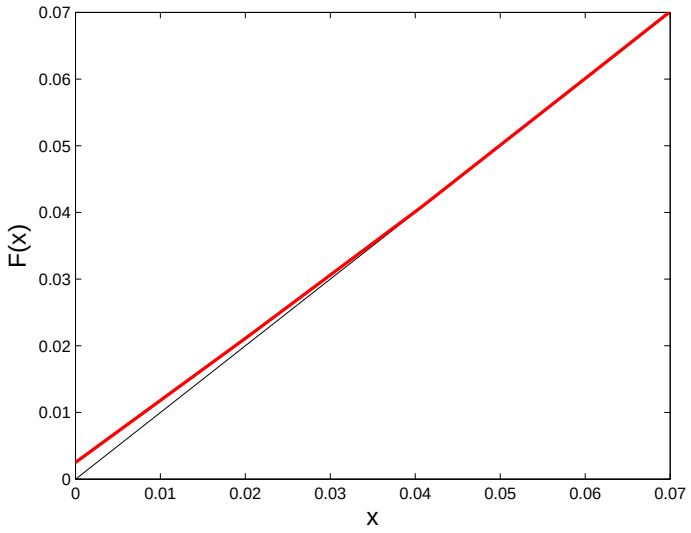


Figure 2.6 – Zoom on $x = 0$ of the EXIT chart of an output degree distribution optimized for a BIAWGN channel of capacity $C = 0.5$. The value in $x = 0$ equals $F(0) = \omega_1 J(2/\sigma^2)$

of the LT code. Note that the total rate of the Raptor code is upper bounded by the channel capacity:

$$R_{LT} R_p < C$$

which, for the setting used in Fig. 2.7, means that the inverse of the rate of the LT code is lower bounded as follows:

$$R_{LT}^{-1} > \frac{R_p}{C} = \frac{0.95}{0.5} = 1.9$$

We also show that there is a minimum value α_{min} under which it is not possible to design *zero error* output degree distributions. Let us first assume that the fountain part of the Tanner graph has converged to its asymptotic value $x_u^{(\infty)} < x_0 < 1$. The extrinsic information content transmitted to the precode is upper bounded by:

$$x_{\text{ext}} \leq J\big(\alpha J^{-1}(x_u^{(\infty)})\big) \tag{2.22}$$

With the re-initialization assumption of the precode Tanner graph (see section 2.2.2), we can assume that the precode is an LDPC code with asymptotic decoding threshold x_p. This means that if the precode is initialized with an information content - coming from the fountain - greater than x_p, then the information content of the precode alone will converge to 1, and the Raptor code has a threshold behavior. It follows that the minimum value of α is given by the condition $x_{\text{ext}} > x_{\text{p}}$, which gives:

$$\alpha \geq \frac{\sigma^2 J^{-1}(x_{\text{p}})}{2} \triangleq \alpha_{min} \tag{2.23}$$

Note that although this condition looks like we implied a tandem decoder, the value of $x_u^{(\infty)}$ is effectively obtained with the joint decoder equations (2.15).

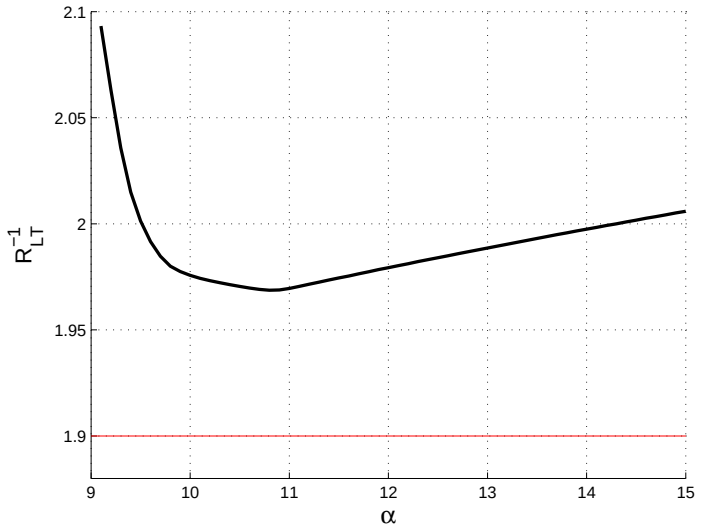


Figure 2.7 – Asymptotic rate of an LT code : R_{LT}^{-1} versus α. For increasing values of α, we optimize a distribution to match a (3,60) regular LDPC precode of rate $R_p = 0.95$ on a BIAWGN channel of capacity $C = 0.5$, and compute the *a posteriori rate* $R_{LT} = \frac{\Omega'(1)}{\alpha}$. It appears that is an optimal value for α that minimizes R_{LT}^{-1} *i.e.* that minimizes the asymptotic overhead.

2.3.3 Parameter δ

The LT part of the Raptor code should be such that at some point of the decoding process, the convergence point reached by the LT code is high enough for the precode to converge, that is to say: x_{ext} becomes larger than the precode threshold x_{p}. For a given value of $\alpha \geq \alpha_{min}$, δ should be such that $J\big(\alpha J^{-1}(x_0 - \delta)\big) \geq x_{\text{p}}$ *i.e.*

$$\delta \leq x_0 - J\left(\frac{\sigma^2 J^{-1}(x_{\text{p}})}{2}\right) \tag{2.24}$$

Fig. 2.8 shows the influence of the parameter δ in the optimization of a distribution. We recall that δ represents a margin away from x_0: the choice $\delta = 0$ leads to an overly stringent optimization problem. Moreover, the larger δ, the higher the achievable asymptotic rate, because the optimization problem becomes less constrained when δ becomes large. However, this inequality shows that δ cannot be chosen arbitrarily, and it should be such that the LT codes reaches a convergence point that is high enough for the precode to converge. In practice, δ can therefore be chosen large, provided that the inequality (2.24) holds.

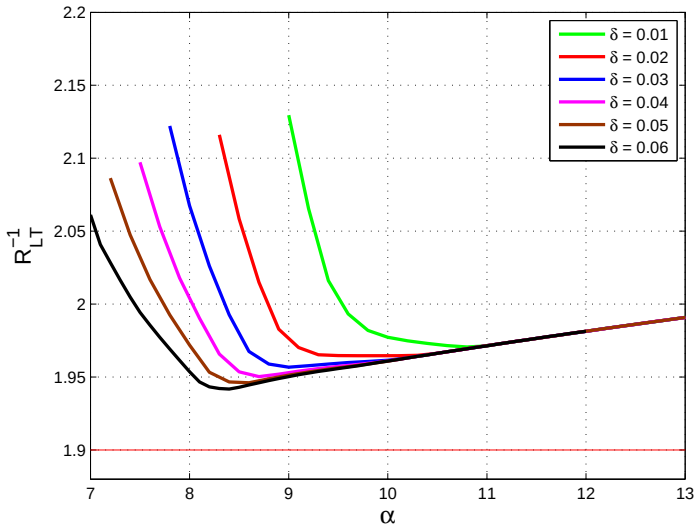


Figure 2.8 – Influence of parameter δ in the optimization of an output degree distribution. For each value of the parameter δ, we optimize a distribution for a BIAWGN channel of capacity $C = 0.5$ with increasing values of α, and compute the rate $R = \frac{\Omega'(1)}{\alpha}$. The plots are for $d_c = 200$.

2.3.4 Parameter d_c

Fig. 2.9 illustrates the influence of the parameter d_c in the optimization. Similarly to the optimization of LDPC codes, it appears that when increasing the maximum degree parameter d_c, it is possible to approach asymptotically closer to the capacity of the underlying channel. However, this comes at the expense of an increased computational decoding complexity, since most of the decoding complexity is related to the update of the outgoing messages of check nodes.

2.3.5 Simulation results

The simulation results are illustrated in terms of BER versus overhead ϵ. We used a regular (3,60) LDPC precode of length $N = 65000$ generated randomly. We compare the distribution $\Omega_E(x)$ proposed in [ES06, p 2044] in both TD and JD schemes, to the following distribution that we optimized for JD with our method:

$$\Omega_B(x) = 0.00428x + 0.49924x^2 + 0.01242x^3 + 0.34367x^4 + 0.04604x^{10} + 0.06181x^{11} + 0.02163x^{22} + 0.01091x^{23} \quad (2.25)$$

For the distribution $\Omega_E(x)$ there is very little difference between TD and JD schemes. This can be explained by the fact that the distribution has not been optimized to take

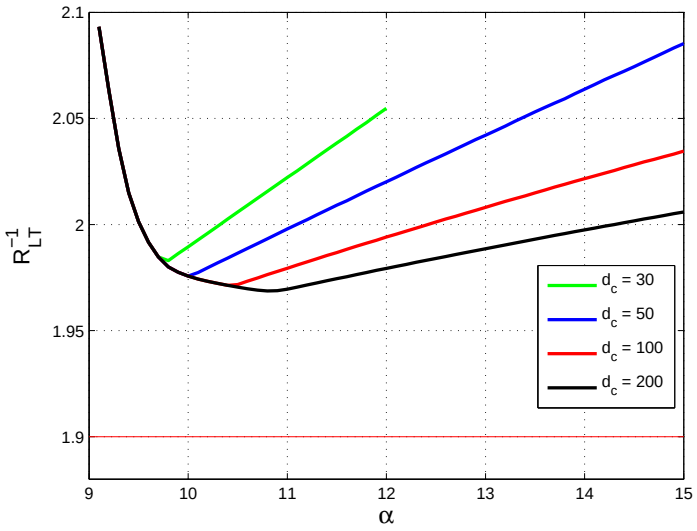


Figure 2.9 – Influence of parameter d_c in the optimization of an output degree distribution, in terms of asymptotic rate: R^{-1} versus α for various values of the maximum degree parameter d_c ($d_c = 10, 30, 50, 100$). The plots are for $\delta = 0.01$.

into account the information provided by the precode. Simulation results are presented in Fig 2.10. Compared to the distribution $\Omega_E(x)$, the distribution $\Omega_B(x)$ appears to operate closer to the channel capacity: the overhead is more that 10% in the first case and less than 5% for our distribution. The thresholds of the corresponding codes are also shown in the figure. This result shows that one can design better output degree distributions by proper optimization with a joint decoder framework.

2.3.6 Robustness against channel parameter mismatch

The results in [ES06] show that Raptor codes are not universal on other channels than the BEC: they cannot adapt to themselves to an unknown channel noise *and* approach the capacity of the channel arbitrarily closely. It was already observed in [PY04] with finite length simulations that Raptor codes designed for the BEC perform well on noisy channels. We show here that a Raptor code that is designed for a given channel performs well on other channels with joint decoding. We compute for different channel capacities the thresholds of the distribution that we optimized for joint decoding, as well as the thresholds of the distribution given in [ES06] with a (3,60) LDPC precode. The results in Fig. 2.11 show that our optimization procedure produces output degree distributions with good thresholds: at $C = 0.4$, the threshold is only 2% away from the capacity of the channel. Fig. 2.12 illustrates the same phenomenon, in terms of achievable rate of a Raptor code for different values of E_s/N_0.

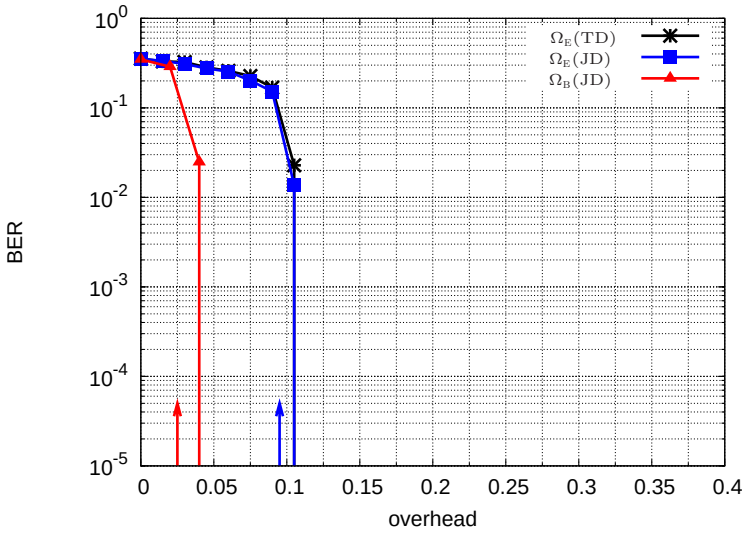


Figure 2.10 – BER versus overhead for a Raptor code defined with a regular (3,60) LDPC precode. We compare $\Omega_B(x)$, a distribution that we optimized for joint decoding, to $\Omega_E(x)$ proposed in [ES06] under tandem decoding (squares) and under joint decoding (stars). The thresholds of the corresponding distributions are indicated with vertical arrows (c.f. section 2.2.6).

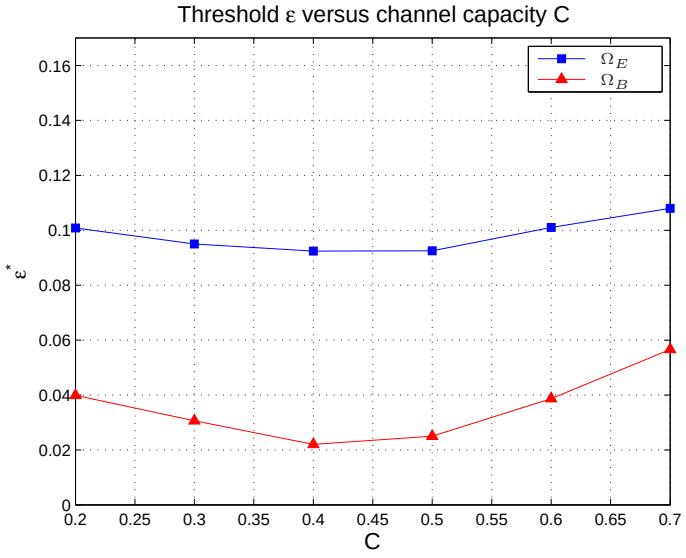


Figure 2.11 – Thresholds of two distributions optimized for $C = 0.5$, for different channel capacities. We compare $\Omega_B(x)$, a distribution that we optimized for joint decoding, to $\Omega_E(x)$ proposed in [ES06] decoded with joint decoding.

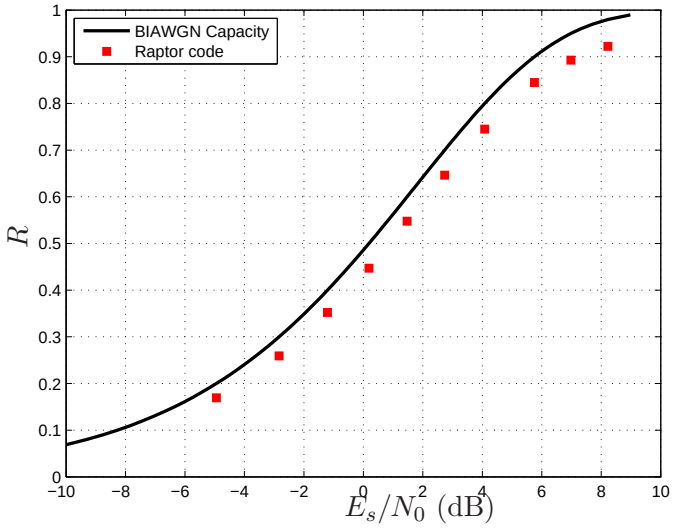


Figure 2.12 – Achievable rates versus E_s/N_0 of a Raptor codes optimized for a BIAWGN channel of capacity $C = 0.8$.

2.4 Finite length design

Finite length performance analysis of LT codes is addressed in [KLS04, MS06a]. In [PS06], the authors propose a model to predict the performance of finite length Raptor codes under iterative decoding, and outline a numerical procedure based on differential evolution to design Raptor codes that perform well at finite length over a Binary Symmetric Channel (BSC).

In this section, we discuss some important points in the perspective of designing efficient Raptor codes for small to moderate lengths. We show that the choice of a rate lower than usually proposed for the precode enables to design Raptor codes that perform well at small lengths, with almost no asymptotic loss. We show in particular that the error floor can be greatly reduced by using a lower rate precode, and that there is almost no asymptotic loss when the output degree distribution is optimized for joint decoding (*i.e.* when the output degree distribution is matched to the precode).

2.4.1 The rate-splitting issue

In the literature, the rate of the precode is usually chosen very close to 1. Indeed, the optimization of output degree distributions allows designing LT codes such that the fraction of unrecovered input symbols is extremely low. Choosing a very high rate precode is a valid strategy when the Raptor code is decoded sequentially, but could be a suboptimal choice when we consider iterative joint decoding of the precode and the LT code. In this latter case, if the output degree distribution is matched – with proper optimization – to the EXIT chart of a *lower rate* precode, there is almost no asymptotic loss *i.e.* no loss in the waterfall region. By *lower rate*, we mean rates that are between $R = 0.9$ and $R = 0.95$, whereas typically in the existing literature, *very high rate* codes, *e.g.* $R = 0.98$, are considered.

Then arises the natural question of the optimal repartition of the overall rate between the LT code and the precode. The use of a lower rate precode can be very attractive in practice, especially for the design of Raptor codes with short or moderate information block lengths. For short to moderate lengths, the topology of the overall Tanner graph in terms of short cycles and subsequent stopping/trapping sets needs to be considered. The problem of using a very high rate LDPC precode is then that it introduces a large number of length-4 cycles, resulting in error floors which are unacceptably high. More precisely, the code length such that an LDPC code of girth 6 exists grows exponentially with the check node degree d_c[HEA05], hence with the code rate. Using a lower rate precode has the main objective of improving the Raptor code in the error floor region for finite block lengths.

We now explain why the use of a lower rate precode does not affect the overall rate of the Raptor code. Let R be the rate of the Raptor code which is the concatenation of

an LT code of rate R_{LT}, and a precode of rate R_p. For a channel with capacity C, the optimization for the sequential decoding scheme always gives $R_{LT} < C$[3]. Since $R_p < 1$, the total rate of the precode $R = R_p R_{LT}$ is smaller than R_{LT}, and therefore the rate of the precode R_p appears to be a burden in terms of the total rate of the Raptor code. However, in the case of the optimization for joint decoding the optimized output degree distributions can have a rate $R_{LT} > C$, which allows considering precodes with lower rates *and* still have a total asymptotic rate R close to the capacity.

Therefore, jointly decoded Raptor codes allow to study the problem of the overall rate distribution and its repartition between the LT code and the precode.

2.4.2 Cycle spectrum of finite length LDPC precodes

For our purpose, we will consider Raptor codes of size $K = 1024, 2048, 4096$ and 8192. We restricted ourselves to regular LDPC precodes because for high rates, regular codes are known to have good thresholds, close to the irregular thresholds. We considered regular LDPC precodes with the following parameters:

- $(d_v, d_c) = (3, 30)$ regular LDPC code of rate $R_p = 0.9$
- $(d_v, d_c) = (3, 40)$ regular LDPC code of rate $R_p = 0.925$
- $(d_v, d_c) = (3, 60)$ regular LDPC code of rate $R_p = 0.95$
- $(d_v, d_c) = (3, 80)$ regular LDPC code of rate $R_p = 0.9625$

The different LDPC precodes (one for each rate and size) were constructed with the RandPEG algorithm that minimizes the multiplicity of the girth, presented in chapter 3. We denote by X-cycle a cycle of length X. All the precodes of size $K = 8192$ are of girth 6 (*i.e.* they have no 4-cycles in their associated Tanner graph). The other (d_v, d_c) LDPC precodes have the following cycle spectrums:

	(d_v,d_c)	code rate (R)	# 4-cycles	# 6-cycles
$K = 4096$	(3,80)	0.9625	643	683392
	(3,60)	0.95	0	259567
	(3,40)	0.925	0	47157
	(3,30)	0.9	0	9728

[3]In the case of the BEC, only the received packets are considered in the decoding graph, and therefore it is convenient to consider that $C = 1$.

	(d_v,d_c)	code rate (R)	# 4-cycles	# 6-cycles
$K = 2048$	(3,80)	0.9625	3536	693044
	(3,60)	0.95	860	289392
	(3,40)	0.925	0	70493
	(3,30)	0.9	0	19982

	(d_v,d_c)	code rate (R)	# 4-cycles	# 6-cycles
$K = 1024$	(3,80)	0.9625	5356	716492
	(3,60)	0.95	2328	295760
	(3,40)	0.925	90	83869
	(3,30)	0.9	0	31394

We emphasize that the 4-cycles codes do not result from a poor construction, but from the fact that for the corresponding rates and sizes, it is not possible to construct regular $(3, d_c)$ LDPC codes [HEA05] of girth 6 (no 4-cycles). To illustrate this fact, the upper bound on the code rate such that a regular $(3, d_c)$ LDPC code of girth 6 and size N exists is shown in Fig. 2.13. The coding rates and sizes of the 16 precodes that we used are also shown in the figure. It appears that our constructions with 4-cycles all correspond to a size and coding rate that do not permit the construction of graphs with no 4-cycles [HEA05].

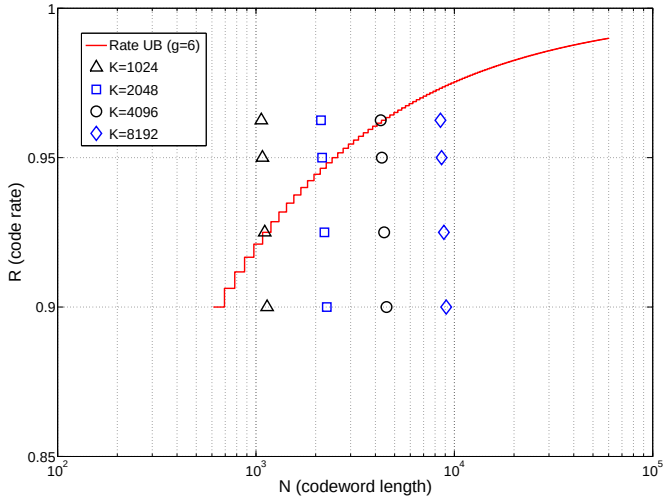


Figure 2.13 – Upper bound on the code rate (Rate UB) such that a regular $(3, d_c)$ LDPC code of girth 6 and size N exists

2.4.3 "Asymptotic design" for finite length distributions

One might question whether the asymptotic analysis of the joint decoder is of any interest for finite length design. In the asymptotic regime, the concentration theorem [RU01] ensures that the performance of a randomly sampled code converges to the expected performance as the codeword length increases. The $x \rightarrow F(x, \sigma^2, T(\cdot))$ characterizes the expected IC evolution of the decoder in the asymptotic regime. In the asymptotic regime, *i.e.* when the codeword length is infinite, the decoding trajectory in the EXIT chart will fit between the curves $y = x$ and $y = F(x, \sigma^2, T(\cdot))$.

However, the concentration to the expected performance does not hold for the finite length case, and one must account for a certain variance in the decoding trajectories. Following the steps of [Sho06], we propose to use the following convergence constraint in the optimization problem for finite length:

[C_2] Convergence constraint:

$$F(x, \sigma^2, T(.)) > x + \frac{c}{K}\sqrt{1-x} \quad \forall x \in [0; x_0 - \delta] \qquad \text{for some } \delta > 0$$

where c is a (small) positive constant.

2.4.4 Simulation results

We optimized output degree distributions for 4 different precodes with different rates. We restricted ourselves to regular LDPC precodes because for high rates, regular codes are known to have good thresholds, close to the irregular thresholds. The optimized output degree distributions are given in Tab. 2.1 to 2.4. Figures 2.14 to 2.17 show simulation results for Raptor codes of length $K = 1024, 2048, 4096$ and 8192 respectively, constructed with precodes described in the previous section.

In fact, according to the cycle spectrum given in section 2.4.2, it appears that all curves that exhibit an error floor are associated with a precode with cycles of length 4. Furthermore, these results show that as long as joint optimization using the precode transfer function is performed, a lower rate precode does not significantly impact the performance of the Raptor code in the waterfall region.

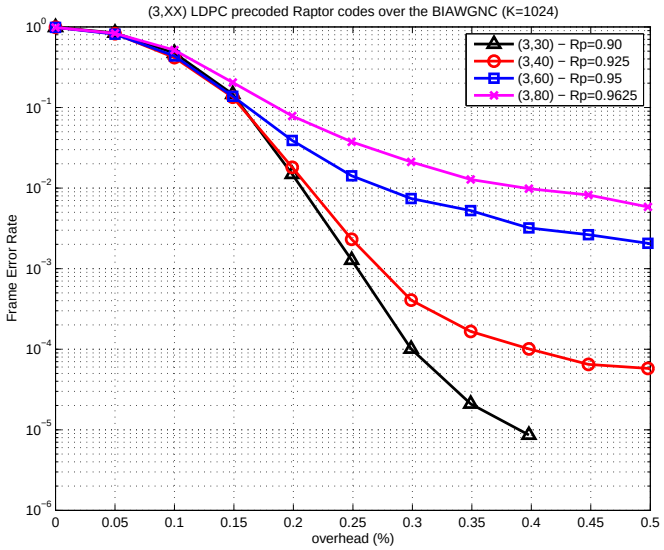


Figure 2.14 – Performance Raptor codes of size $K = 1024$ input bits over a BIAWGN channel of capacity $C = 0.5$, for different precode rates.

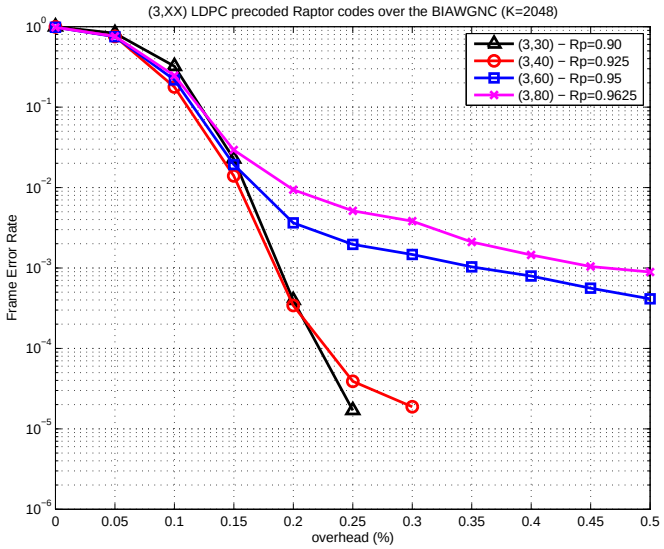


Figure 2.15 – Performance Raptor codes of size $K = 2048$ input bits over a BIAWGN channel of capacity $C = 0.5$, for different precode rates.

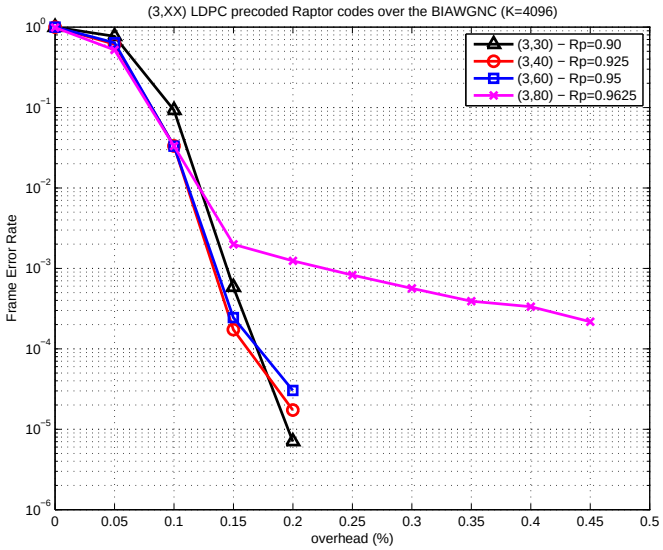


Figure 2.16 – Performance Raptor codes of size $K = 4096$ input bits over a BIAWGN channel of capacity $C = 0.5$, for different precode rates.

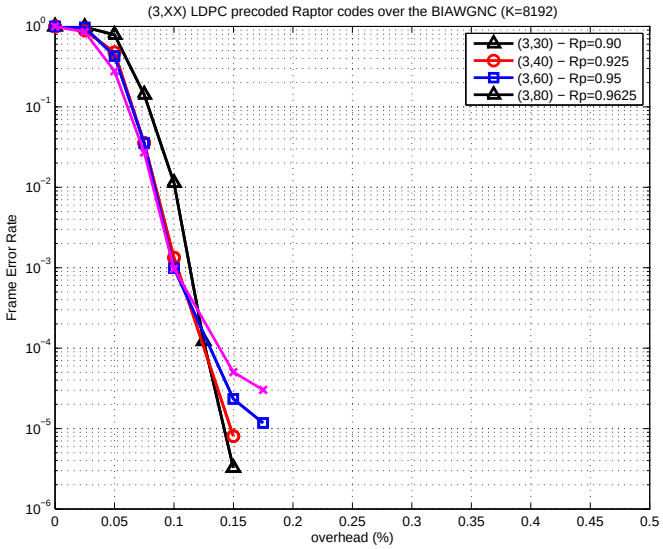


Figure 2.17 – Performance Raptor codes of size $K = 8192$ input bits over a BIAWGN channel of capacity $C = 0.5$, for different precode rates.

$\Omega(x)$	$(3,30)$	$(3,40)$	$(3,60)$	$(3,80)$
Ω_1	0.05501	0.05742	0.06449	0.06600
Ω_2	0.49500	0.48435	0.45750	0.45103
Ω_3	0.04632	0.02912	0.06250	0.06770
Ω_4	0.26061	0.29748	0.20572	0.19956
Ω_5			0.09380	0.09319
Ω_9	0.13613			
Ω_{10}	0.00693	0.07668		
Ω_{11}		0.05495		
Ω_{14}			0.07155	0.04902
Ω_{15}			0.04444	0.07350
$\Omega'(1)$	3.52088	3.67465	4.12717	4.22413

Table 2.1 – Output degree distributions optimized for regular $(3,30)$, $(3,40)$,$(3,60)$ and $(3,80)$ LDPC precodes of size $K = 1024$. $\Omega'(1)$ is the average degree of an output symbol.

$\Omega(x)$	$(3,30)$	$(3,40)$	$(3,60)$	$(3,80)$
Ω_1	0.03961	0.04133	0.04645	0.04762
Ω_2	0.50402	0.49301	0.46664	0.46020
Ω_3	0.02891	0.02055	0.05119	0.06513
Ω_4	0.29747	0.31883	0.27663	0.24480
Ω_6				0.05981
Ω_7			0.06023	0.01283
Ω_9	0.03914			
Ω_{10}	0.09085	0.01295		
Ω_{11}		0.11333		
Ω_{15}			0.03891	0.03517
Ω_{16}			0.05995	0.07444
$\Omega'(1)$	3.58502	3.74045	4.20428	4.30987

Table 2.2 – Output degree distributions optimized for regular $(3,30)$, $(3,40)$,$(3,60)$ and $(3,80)$ LDPC precodes of size $K = 2048$. $\Omega'(1)$ is the average degree of an output symbol.

$\Omega(x)$	$(3,30)$	$(3,40)$	$(3,60)$	$(3,80)$
Ω_1	0.02837	0.02959	0.03331	0.03415
Ω_2	0.51053	0.49922	0.47330	0.46681
Ω_3	0.02159	0.01733	0.06467	0.06265
Ω_4	0.31599	0.32980	0.25761	0.26435
Ω_7			0.05951	0.07194
Ω_8			0.02819	
Ω_{10}	0.10546			
Ω_{11}	0.01806	0.10026		
Ω_{12}		0.02380		
Ω_{16}				0.04643
Ω_{17}			0.08341	0.05367
$\Omega'(1)$	3.63142	3.78768	4.26442	4.37197

Table 2.3 – Output degree distributions optimized for regular $(3,30)$, $(3,40)$,$(3,60)$ and $(3,80)$ LDPC precodes of size $K = 4096$. $\Omega'(1)$ is the average degree of an output symbol.

$\Omega(x)$	$(3,30)$	$(3,40)$	$(3,60)$	$(3,80)$
Ω_1	0.02024	0.02111	0.02379	0.02440
Ω_2	0.51517	0.50360	0.47807	0.47158
Ω_3	0.01430	0.01536	0.06111	0.07156
Ω_4	0.33116	0.33742	0.27302	0.25150
Ω_7				0.08631
Ω_8			0.08397	
Ω_9			0.00774	
Ω_{10}	0.06419			
Ω_{11}	0.05494	0.07331		
Ω_{12}		0.04920		
Ω_{17}				0.07981
Ω_{18}			0.06297	0.01484
Ω_{19}			0.00933	
$\Omega'(1)$	3.66436	3.82088	4.30749	4.4163

Table 2.4 – Output degree distributions optimized for regular $(3,30)$, $(3,40)$,$(3,60)$ and $(3,80)$ LDPC precodes of size $K = 8192$. $\Omega'(1)$ is the average degree of an output symbol.

2.5 New results on LT codes and Raptor codes under tandem decoding

An LT code can be seen as a Raptor code with no precode. Thus, the convergence analysis and optimization method described in the previous section allow to design LT codes by assuming that $T(x) = 0 \quad \forall x \in [0; 1]$ in the IC evolution equations. First, we will show that there is a trade-off in the design between the asymptotic error floor and the achievable rate of an LT code, and that this trade-off can be directly controlled with the design parameter α. For that purpose, we derive upper and lower bounds for the average bit error probability in the error floor region. Then we present a model for the equivalent channel seen by the precode in the case of the tandem decoding scheme and derive an optimization method for the design of a precode.

2.5.1 Asymptotic error floor of an LT code

In the case of the BEC, the fact that the output degree distribution has a constant average degree (not growing with the code length) implies that there will be a fraction of unrecovered input symbols. In the case of the BIAWGN channel, it implies that a fraction of input symbols will not be well estimated: an error floor phenomenon is observed. In that context, our purpose in this section is to characterize the performance of the output degree distributions in terms of residual bit error rate (BER), which we define as the BER after an infinite number of iterations.

2.5.1.1 Characterization of the error floor region

Similarly to [CRU01], and assuming that the decoding process has reached its fixed point $x_u^{(\infty)}$, the error probability for an input symbol of degree i is given by:

$$P^i = Q\left(\sqrt{\frac{iJ^{-1}(x_u^{(\infty)})}{2}}\right) \tag{2.26}$$

where $Q(\cdot)$ is the Gaussian complementary cumulative distribution function. The average error probability P_e is obtained by averaging over the input symbol degree distribution $I(x)$:

$$P_e = \sum_i I_i P^i$$

The residual probability of error is a decreasing function of α, because $I(x)$ is a Poisson distribution with mean α. Thus, α appears to be a design parameter for controlling the residual BER of an LT code. However, Fig. 2.7 shows that there is a value for α that gives the best asymptotic performance. Therefore, a trade-off appears in the design of

an LT code: on the one hand, the value that gives minimizes the overhead corresponds to a certain BER performance, but on the other hand, it is possible to have a lower residual BER by increasing α, at the price of a higher asymptotic overhead.

Recalling that $x_u^{(\infty)} > x_0 - \delta$ because of the convergence constraint and that $x_u^{(\infty)} < x_0$ (2.17), we get respectively an upper bound and a lower on the residual BER of the LT code, given by:

$$\sum_i I_i Q\left(\sqrt{\frac{iJ^{-1}(x_0)}{2}}\right) < P_e < \sum_i I_i Q\left(\sqrt{\frac{iJ^{-1}(x_0-\delta)}{2}}\right) \tag{2.27}$$

For an output degree distribution $\Omega(x)$, the asymptotic rate is related to an asymptotic overhead under Gaussian approximation by the following: $R_{LT}(1+\epsilon^*) = C$. Thus, the design parameter α is also related to an asymptotic overhead ϵ^*:

$$\alpha = \frac{\Omega'(1)}{R_{LT}} = \frac{(1+\epsilon^*)\Omega'(1)}{C}$$

Then, for an overhead $\epsilon > \epsilon^*$, the actual average input degree $\tilde{\alpha}$ is given by:

$$\tilde{\alpha} = \frac{(1+\epsilon)\Omega'(1)}{C}$$

Replacing $I(x)$ by a Poisson distribution with parameter $\tilde{\alpha} = f(\epsilon)$ in eq (2.27) gives the upper and lower bounds as a function of the overhead.

2.5.1.2 Simulation results

We have designed three output degree distributions, for different values of α. The resulting output degree distributions are reported in Table 2.5, and Fig. 2.18 shows the BER versus overhead for these distributions. The value $\alpha = 14$ corresponds to the lowest asymptotic overhead. With larger values for α, we designed distributions with larger overhead but lower error floors, as predicted in the previous section. For each distribution, the upper and lower bounds are also shown in the figure, and although they are based on a Gaussian approximation, it appears that they are quite tight, and thus, they provide a good tool for predicting the BER of an LT code in the error floor region.

2.5.2 Design of a precode for tandem decoding

In this section, we consider the tandem decoding scheme, and propose an optimization method for the left irregularity profile of an LDPC precode, based on IC evolution under Gaussian approximation. The optimization method is based on the fact that, since the input symbol node degree distribution is well approximated by a Poisson distribution, the channel seen by the precode can be modeled by a mixture of Gaussian channels of different capacities.

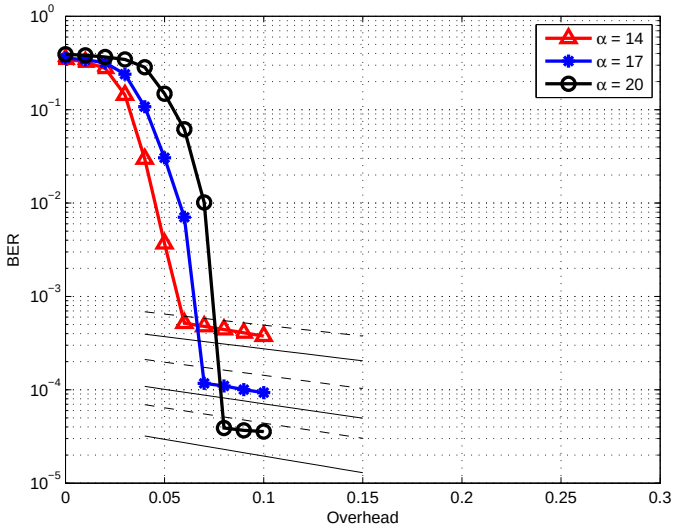


Figure 2.18 – BER versus overhead for LT codes designed with different values of the parameter α. The upper and lower bounds for the BER (section 2.5.1) plotted in the figure appear to be quite tight. All simulations were run for $K = 65000$ input symbols, on a BIAWGN channel of capacity $C = 0.5$ ($\sigma = 0.9787$) with 600 decoding iterations.

2.5.2.1 IC evolution for a mixture of Gaussian channels

First, we present the optimization of an LDPC code for a channel that can be modeled by a mixture of N_c Gaussian channels with known Channel State Information (CSI) at the receiver.

We denote N_c the number of different channels involved in the mixture, σ_m the channel parameter associated with the m^{th} channel, and p_m the probability associated with channel parameter σ_m^2 in the mixture. Moreover, let $\lambda(x)$ (resp. $\rho(x)$) be the variable (resp. check) edge degree distribution, and let $x_{up}^{(l)}$ (resp. $x_{vp}^{(l)}$) denote the IC associated to the messages on an edge connecting a check node to a variable node (resp. a variable node to a check node) at the l^{th} decoding iteration of the precode. Using IC evolution, the BP update equations are given by:

- Variable node message update:

$$x_{vp}^{(l)} = \sum_{i=2}^{d_v} \lambda_i \sum_{m=1}^{N_c} p_m J\Big(\frac{2}{\sigma_m^2} + (i-1)J^{-1}(x_{up}^{(l-1)})\Big) \tag{2.28}$$

- Check node message update:

$$x_{up}^{(l)} = 1 - \sum_{j=2}^{d_c} \rho_j J\Big((j-1)J^{-1}(1-x_{vp}^{(l)})\Big) \tag{2.29}$$

	α		
$\Omega(x)$	14	17	20
Ω_1	0.00683	0.00825	0.00481
Ω_2	0.44724	0.43883	0.43063
Ω_3	0.07537	0.08077	0.11258
Ω_4	0.27843	0.28034	0.24300
Ω_9	0.04518	0.00194	0.08654
Ω_{10}	0.06949	0.11672	0.02985
Ω_{16}			0.03116
Ω_{21}		0.03050	
Ω_{22}	0.02943		
Ω_{23}	0.01768		
Ω_{35}			0.01833
Ω_{36}			0.02074
Ω_{44}		0.02791	
Ω_{80}	0.02404		
Ω_{81}	0.00633		
Ω_{200}		0.01474	0.02236
$\Omega'(1)$	6.8311	8.2508	9.3119

Table 2.5 – Coefficients of the output degree distributions of LT codes optimized with different values of α. $\Omega'(1)$ is the average degree of an output symbol.

Replacing (2.29) in (2.28) gives the equation that describes the evolution through one decoding iteration of the IC of the LDRs at the output of the check nodes of the precode, given $\left(\{p_m, \sigma_m^2\}\right)_{m\in[1:N_c]}$, which gives the monodimensionnal recursive equation: $x_{vp}^{(l)} = F_p(x_{vp}^{(l-1)})$. Note that for a given distribution $\rho(x)$, this expression is linear with respect to the coefficients of $\lambda(x)$, which is the distribution that we intend to optimize.

As a straightforward generalization of the Gaussian case, the stability condition is given by:

$$\lambda_2 \sum_{m=1}^{N_c} p_m e^{-\frac{1}{2\sigma_m^2}} < \frac{1}{\sum_{j=2}^{d_c} \rho_j (j-1)} \tag{2.30}$$

2.5.2.2 Application to the design of a precode

This model can be used for the design of a precode for Raptor codes. If we denote x_f the IC of the LDR messages on the edges connecting the dynamic check nodes to the input symbols in the LT code, then the channel parameters σ_m^2 are directly related to x_f by the following relation:

$$\frac{2}{\sigma_m^2} = mJ^{-1}(x_f) \tag{2.31}$$

where m is the degree of an input symbol of the LT code. It follows that the N_c channel parameters are a function of the degrees m and a "global channel parameter" x_f. Moreover, due to the distribution of the input symbols of the LT code, p_m is Poisson distributed.

For a given x_f and $\rho(x)$, the optimization problem can finally be stated as follows:

$$\lambda_{opt}(x) \quad = \quad \arg\min_{\lambda(x)} \sum_{i=2}^{d_v} \frac{\lambda_i}{i} \tag{2.32}$$

subject to the constraints:

[C_1] Proportion constraint : $\sum_i \lambda_i = 1$

[C_2] Convergence constraints: $F_p(x) > x \quad \forall x \in [0; 1]$

[C_3] Stability condition: see eq. (2.30)

For a concentrated $\rho(x)$ distribution [CRU01], the optimization is finally done by threshold maximisation, through the equivalent channel parameter x_f: we intend to minimize the parameter x_f that allows convergence of the precode for a given target rate. Indeed, because of eq (2.31), minimizing x_f is equivalent to maximizing the channel noise parameter.

2.5.2.3 Simulation results

The same distribution is tested with two different precodes of rate $R = 0.95$, with tandem decoding on a BIAWGN channel of capacity $C = 0.5$. The first precode is optimized for a mixture of Gaussian channels, and the second is the code with the best threshold on a BIAWGN channel, optimized via density evolution [Lop]. The simulation results in Fig. 2.19 show that for a given distribution, the optimization of the precode allows to operate closer to the channel capacity. We used tandem decoding with 600 decoding iterations for the LT code and 600 decoding iterations for the precode.

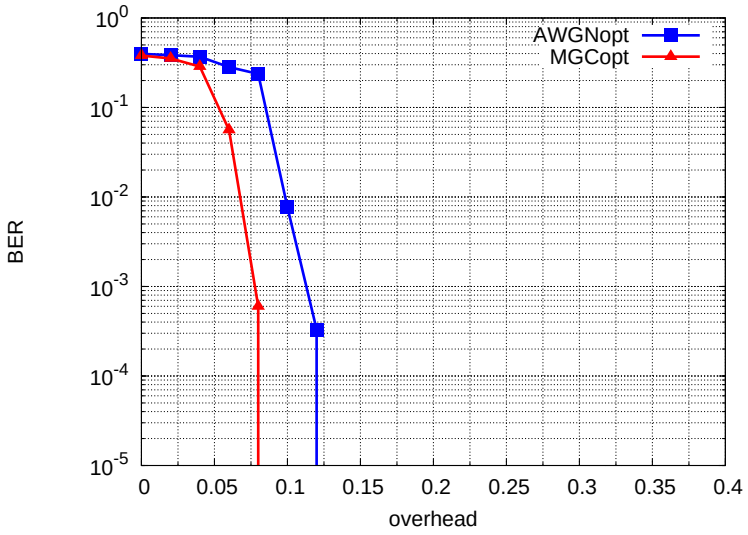


Figure 2.19 – The same distribution (Table 2.5, $\alpha = 14$) decoded with classical tandem decoding, with *(i)* a precode optimized for a mixture of Gaussian channels (MGCopt), versus *(ii)* the LDPC code on [Lop] of rate 0.95, with the best threshold on a BIAWGN channel (AWGNopt). Simulations where run on a BIAWGN channel of capacity $C = 0.5$ with $K = 65000$ input symbols. .

2.6 Analysis of Raptor codes on uncorrelated fading channels

In this section, we propose an asymptotic analysis of jointly decoded Raptor codes for an uncorrelated Rayleigh fading channel, also called fast Rayleigh fading channel. We assume that CSI is available at the receiver but not at the transmitter. For the analysis, we use IC evolution under Gaussian approximation. We show that with only a single minor modification, the IC evolution equations for the BIAWGN channel remain valid for the uncorrelated Rayleigh fading channel.

2.6.1 Channel model

We consider the uncorrelated flat Rayleigh fading channel, with perfect Channel State Information at the Receiver (CSIR). This channel model is justified as follows. In a wireless transmission, various phenomena such as multipath propagation, terminal mobility and users interference, result in channels with time-varying parameters. One important parameter is the channel coherence time, defined as the number of channel uses where the channel can be considered as non varying. The coherence time is related to the Doppler spread, and depends on many parameters such as the carrier frequency, communication bandwidth (see e.g. [TV05]).

A realistic wireless communication model is the block fading model: during a communication, the receiver will experience several channel conditions, corresponding to different blocks of length the coherence time, over which one may assume that the receiver can estimate the channel parameter with pilot based estimation techniques for example. This justifies that CSIR is available.

In a Raptor based communication scheme, all the output symbols are independent and therefore it does not matter from which block they come from. Consequently, all the received output symbols can be randomly permuted, provided that the graph setting (*i.e.* information about how output symbols and input symbols are connected) is preserved. If the number of transmitted blocks is sufficient (typically a few hundreds), then block fading transmission with fully interleaved coded transmission can be modeled by an uncorrelated fast fading transmission with CSIR.

Let us now review the channel model. At the output of the channel, we have:

$$Y_i = a_i X_i + N_i$$

The input symbols X_i belong to a BPSK modulation and the energy per symbol is normalized so that $\mathbb{E}_X\{|X_i|^2\} = 1$, where $\mathbb{E}\{\cdot\}$ denotes expectation, and N_i is a Gaussian noise with variance σ_b^2. We consider real inputs/outputs at the channel because a perfect demodulation is possible when *perfect* CSIR is available.

Furthermore, the channel fading is a Rayleigh distributed random variable A normalized

such that $\mathbb{E}[a^2] = 1$, with the following pdf:

$$\mathrm{p(a)} = 2a \exp\left(-a^2\right)$$

When perfect CSIR is assumed, the *a priori* pdf based on the observations is given by:

$$\mathrm{p(y|x,a)} = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{|y-ax|^2}{2\sigma_b^2}\right)$$

The log-likelihood ratios (LLR), associated with this pdf, are computed as follows:

$$L_M(y) = \frac{2}{\sigma_b^2} ay \tag{2.33}$$

Conditioned to a and x, $L_M(y)$ is a consistent Gaussian random variable [HSM01] with pdf $\mathcal{N}(\frac{2a^2}{\sigma_b^2}, \frac{4a^2}{\sigma_b^2})$. Thus, the information content of LLRs associated with the fading coefficient a is given by $x_0(a) = J(m_0(a))$ with $m_0(a) = \frac{2a^2}{\sigma_b^2}$.

The signal to noise ratio in dB is given by

$$\frac{E_s}{N_0}_{dB} \triangleq 10 \log_{10}\left(\frac{2}{\sigma_b^2}\right)$$

2.6.2 IC evolution for uncorrelated Rayleigh fading channels

We use the same notations as in section 2.2 for the IC evolution equations. We will show in this section that the IC evolution equations of the analysis for the BIAWGN channel remain valid for the uncorrelated Rayleigh fading channel provided that the constant f_0, that depends on the channel parameter σ_b^2 in the BIAWGN case, is properly redefined.

For a given channel realization a, we have at the output of a dynamic check node of degree j:

$$J^{-1}(1 - x_v^{(l)}) = (j-1)J^{-1}(1 - x_u^{(l-1)}) + J^{-1}\left(1 - J\left(\frac{2a^2}{\sigma_b^2}\right)\right) \tag{2.34}$$

Which gives:

$$x_v^{(l)} = 1 - J\left((j-1)J^{-1}(1 - x_u^{(l-1)}) + J^{-1}\left(1 - J\left(\frac{2a^2}{\sigma_b^2}\right)\right)\right) \tag{2.35}$$

Since the channel gain a is a Rayleigh distributed variable, the expected value of the IC at the output of a dynamic check node is computed as follows:

$$x_v^{(l)} = 1 - \int_{[0;\infty]} J\left((j-1)J^{-1}(1 - x_u^{(l-1)}) + J^{-1}\left(1 - J\left(\frac{2a^2}{\sigma_b^2}\right)\right)\right) 2ae^{-a^2} da \tag{2.36}$$

Eq. (2.36) involves a computationally expensive integration. The integral depends on the check node degree, the channel noise parameter, and the IC x_u. Therefore, contrary to the integral in eq. (2.8) that defines $J(\cdot)$ with m as a single parameter, it is a computational burden to tabulate the corresponding integral for a Rayleigh fading channel.

We propose to use the following approximation:

$$x_v^{(l)} = 1 - J\Bigg((j-1)J^{-1}(1-x_u^{(l-1)}) + J^{-1}\Big(1 - \int_{[0;\infty]} J\Big(\frac{2a^2}{\sigma_b^2}\Big)2ae^{-a^2}da\Big)\Bigg) \tag{2.37}$$

In order to test the validity of both approaches, we have tested for various check node degrees the validity of eq. (2.36) and its approximation eq. (2.37) by simulating the output of a dynamic check node. We used the following Monte Carlo approach to simulate the output of an uncorrelated Rayleigh fading channel. First, the fading coefficients are randomly sampled according to a normalized Rayleigh distribution. Then the LLR messages, that are symmetric and Gaussian distributed [HSM01], and where sampled according to the following law: $\mathcal{N}(\frac{2a^2}{\sigma_b^2}, \frac{4a^2}{\sigma_b^2})$

The messages from variable nodes to a dynamic check node were sampled according to a Gaussian symmetric distribution, and the output of the dynamic check node is computed according to the BP update rule.

We used the following estimate for the IC at the output of a dynamic check node: Let $(m_k)_{k=1:N}$ denote N samples of a random variable. Let x_{MC} denote the IC associated to m, estimated with Monte Carlo:

$$x_{MC}(m_k) = 1 - \frac{1}{N}\sum_{k=1:N} \log_2(1+\exp(-m_k)) \tag{2.38}$$

We present some simulation results in Fig. 2.20 to 2.21. The two plots corresponding to eq. (2.36) and eq. (2.37) match perfectly, which indicates that eq. (2.37) can be indeed used as an approximation of eq. (2.36). We do not report any results for the output of a variable node, because the Gaussian approximation is already known to be accurate at the output of a variable node [AK04].

We point out that

$$C_{Rayleigh} = \int_{[0;\infty]} J\Big(\frac{2a^2}{\sigma_b^2}\Big)2ae^{-a^2}da$$

is the ergodic capacity of an uncorrelated Rayleigh fading channel. In the analysis for the BIAWGN channel, the channel model appears through the channel capacity in the constant f_0. Therefore the study in this section shows that the important quantity in the IC evolution equations is the channel ergodic capacity. It follows that the analysis for a BIAWGN channel remains valid for the uncorrelated Rayleigh fading channel,

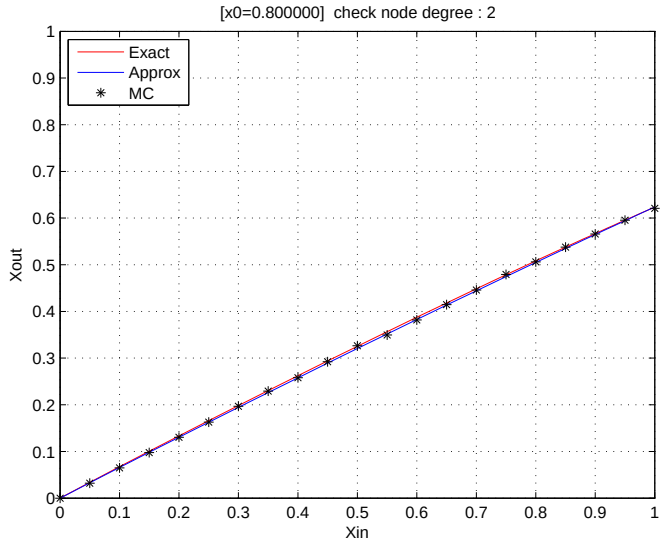


Figure 2.20 – IC evolution at a dynamic check node of degree $j = 2$

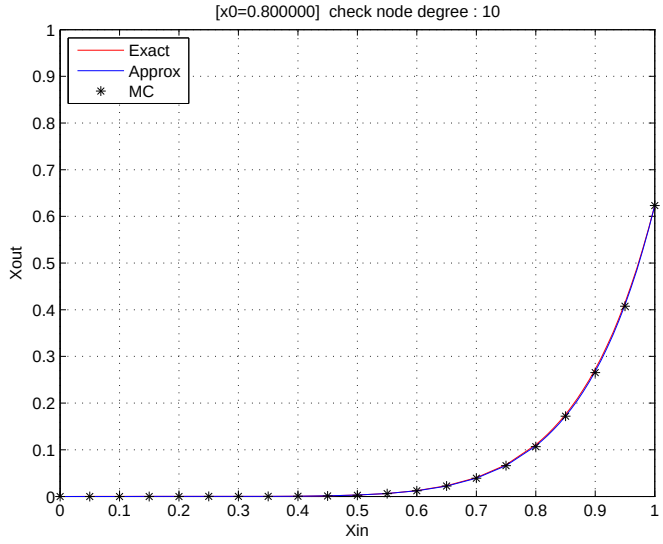


Figure 2.21 – IC evolution at a dynamic check node of degree $j = 10$

provided that the constant f_0 is redefined as follows:

$$\begin{aligned} f_0 &\triangleq J^{-1}\Big(1 - \int_{[0;\infty]} J\Big(\frac{2a^2}{\sigma_b^2}\Big) 2ae^{-a^2} da\Big) \\ &= J^{-1}\Big(1 - C_{Rayleigh}\Big) \end{aligned}$$

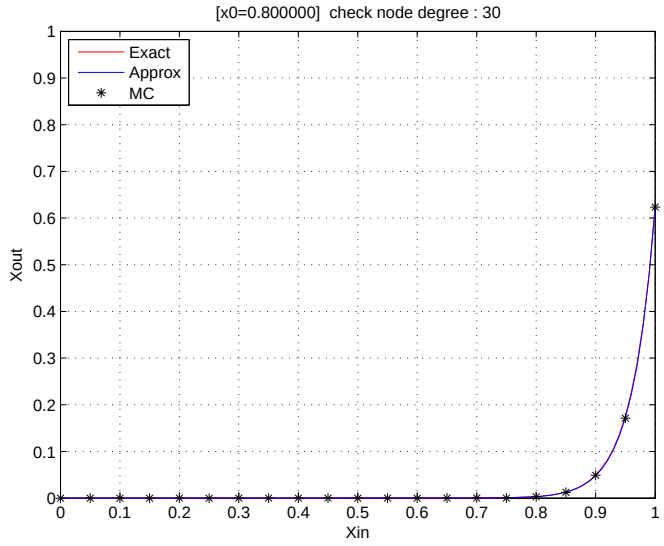


Figure 2.22 – IC evolution at a dynamic check node of degree $j = 30$

2.6.3 Simulation results

First we show that output degree distributions optimized for a given channel capacity perform well over a wide range of channel capacities. Fig. 2.23 shows the asymptotic overheads estimated with Algo. 2 of the output degree distribution optimized in section 2.3.6. The asymptotic overhead is less that 6% for capacities ranging from 0.1 to 0.9.

In Fig. 2.24 to 2.27 , we present finite length simulations are presented for an uncorrelated Rayleigh fading channel (BPSK input) of capacity $C = 0.5$ (Normalized fading, $\sigma_b = 0.81000$). The simulation results are very similar to the simulations on the BI-AWGN channel (Fig. 2.14 to 2.17) and confirm that the optimization for the BIAWGN channel holds for the uncorrelated Rayleigh fading channel, as predicted with the IC evolution analysis.

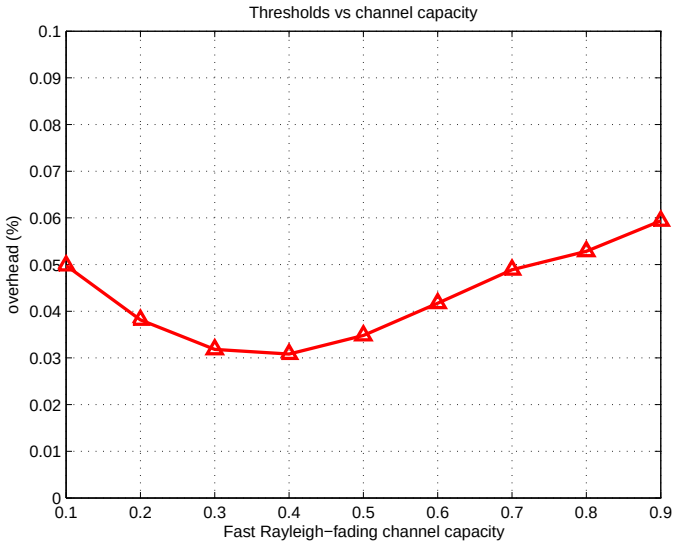


Figure 2.23 – Overhead thresholds on uncorrelated Rayleigh fading channel. The distribution is optimized for a channel capacity $C = 0.5$ and a (3,60) regular LDPC precode. The thresholds on fast Rayleigh fading channels is evaluated with Algorithm 2

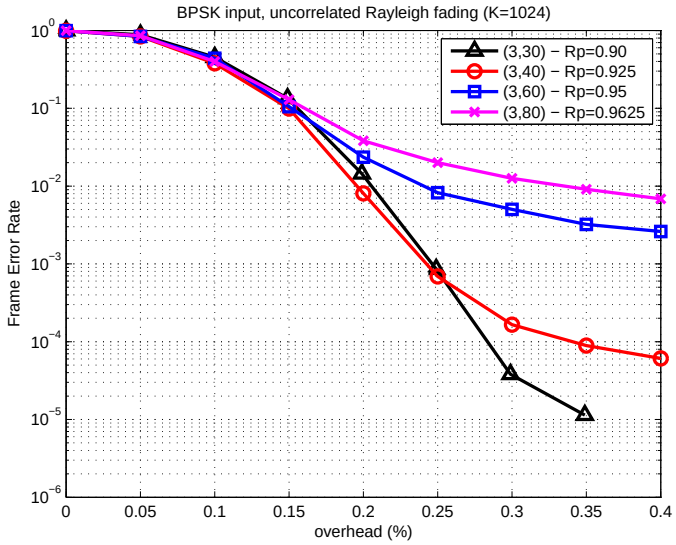


Figure 2.24 – Performance, in terms of FER versus overhead, of Raptor codes of size $K = 1024$ over an uncorrelated Rayleigh fading channel with BPSK input. The channel is of capacity $C = 0.5$

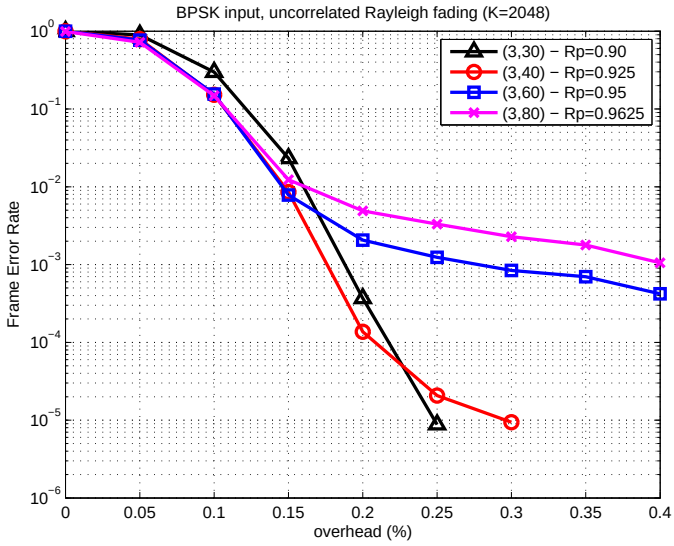


Figure 2.25 – Performance, in terms of FER versus overhead, of Raptor codes of size $K = 2048$ over an uncorrelated Rayleigh fading channel with BPSK input. The channel is of capacity $C = 0.5$

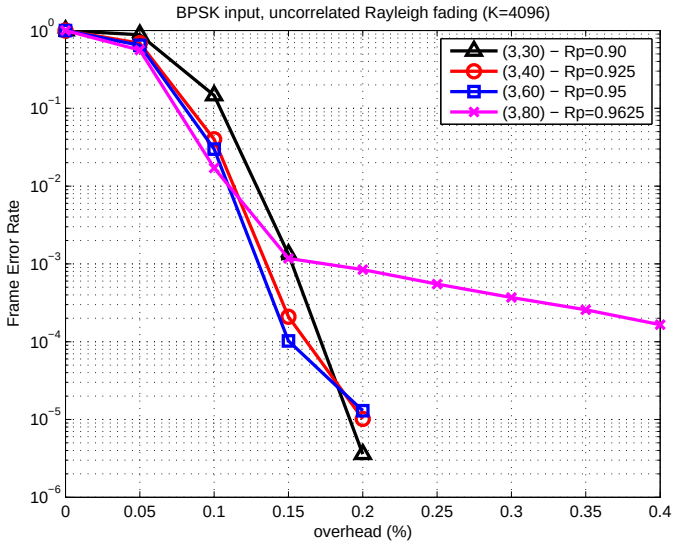


Figure 2.26 – Performance, in terms of FER versus overhead, of Raptor codes of size $K = 4096$ over an uncorrelated Rayleigh fading channel with BPSK input. The channel is of capacity $C = 0.5$

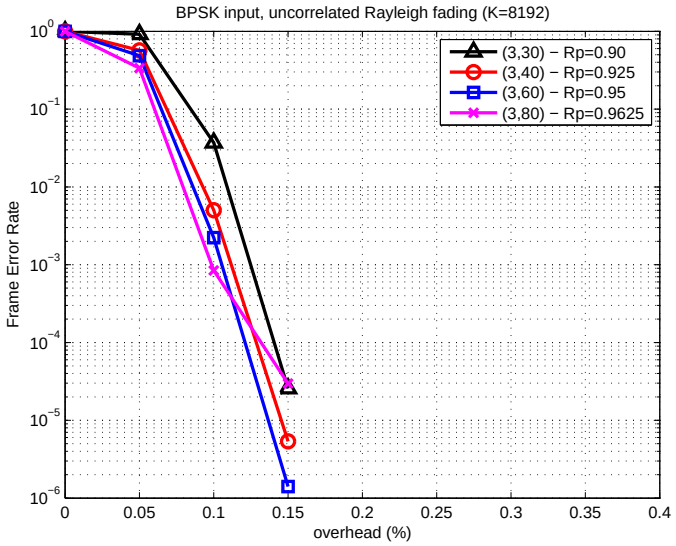


Figure 2.27 – Performance, in terms of FER versus overhead, of Raptor codes of size $K = 8192$ over an uncorrelated Rayleigh fading channel with BPSK input. The channel is of capacity $C = 0.5$

2.7 Analysis of Raptor codes on quasi-static fading channels

In [SVW06, SVW05], the authors compare an IR-HARQ scheme based on punctured LDPC codes, and a HARQ scheme based on Raptor codes [Sho06] for communication over time-varying channels. They characterized the Maximum-Likelihood (ML) performance of the two schemes by exhibiting for each one, bounds on the asymptotic left tail of the code spectrum. They do not directly address the optimization of Raptor codes, but rather their approach relies on a power allocation strategy. In [CM06b, CM06a], the use of rateless codes for communication over Rayleigh fading channel is addressed. Nevertheless, the authors use codes optimized for a BIAWGN channel of capacity $C = 0.5$, resulting in a wide gap to the channel capacity in the high SNR regime.

In this section, we show how Raptor codes can be optimized for communication over quasi-static fading channels under delay constraints, when CSI is available at the receiver but not at the transmitter. To that end, we extend the application of rateless coding to non-ergodic memoryless channels. In particular, we investigate the capability of Raptor codes to select their decoding time dynamically to match the rate of communication to the instantaneous channel capacity. We first introduce rateless coding for non-ergodic channels and characterizes its ultimate performances. Based on the notion of outage capacity we characterize the optimal trade-off between the coding rate and the decoding time. Then, we propose an optimization method for Raptor codes on quasi-static fading channels. Simulation results show that such codes efficiently adapt in an opportunistic manner to the communication conditions, *i.e.*, decoding in less time when the channel is good and taking more time to decode when the channel is bad.

2.7.1 Channel model

From now on, we assume that the channel state, which is unknown at the transmitter is constant within the transmission block. We consider perfect channel state information at the receiver (CSIR); the case of imperfect CSIR is addressed in section 2.8.3. At the output of the channel, we have:

$$Y_i = aX_i + N_i$$

The input symbols X_i belong to a BPSK modulation and the energy per symbol is normalized so that $\mathbb{E}_X\{|X_i|^2\} = P$, where $\mathbb{E}\{\cdot\}$ denotes expectation. We consider real inputs/outputs at the channel because a perfect demodulation is possible when *perfect* CSIR is available.

Furthermore, the channel fading is a Rayleigh distributed random variable A normalized such that $\mathbb{E}[a^2] = 1$, with the following pdf:

$$\mathrm{p(a)} = 2a \exp\left(-a^2\right)$$

2.7.2 The rateless paradigm for non-ergodic channels

Conditioned on the channel realization $A = a$, the instantaneous channel is a BIAWGN channel with capacity

$$C(a) = \max_{P_X} I(X; Y, A = a). \tag{2.39}$$

Therefore, a quasi-static fading channel can be seen as an AWGN channel where the capacity $C(a)$ is a random variable. In the rateless setting, the inverse of the information rate is proportional to the delay needed to correctly decode the information sent by the encoder. Hence, the number of bits required to decode such code only depends on the current channel draw, *i.e.* decoding in less time when the instantaneous channel realization is good and taking more time to decode when the channel is bad. Thus, a decoder might not support the maximal information delay Δ tolerated and an outage will occur with a certain probability.

We consider outage events induced by information delays (*i.e.* decoding time). Note that with infinite delay, rateless codes guarantees no outage events. This follows from the fact that if at some point the receiver has not recovered the transmitted message, we do not consider that an outage occurs, but instead that the receiver must collect more data in order to recover the message. In the sequel, we will not consider the coding rate R, rather the delay given by its inverse $\Delta = R^{-1}$.

2.7.3 Theoretical limits

We propose to measure the theoretical performances of rateless codes in terms of the probability of decoding delay $P\text{wait}(\Delta)$, defined as the probability that the information sent by the encoder at a given delay Δ (or rate $R = \Delta^{-1}$) be not sufficient for the instantaneous channel realization, and that the receiver must wait for more data.

$$P_{\text{wait}}(\Delta) = \Pr\{I(X; Y, A = a) < \Delta^{-1}\}.$$

Given a system requirement for the delay probability p_{wait}, we define Δ^* as the minimum delay such that for any delay $\Delta \geq \Delta^*$ ($\geq \Delta_{\min}$ delay minimal) the probability $P_{\text{wait}}(\Delta) \leq p_{\text{wait}}$,

$$\Delta^*(p_{\text{wait}}) = \inf\{\Delta \geq \Delta_{\min} : P_{\text{wait}}(\Delta) \leq p_{\text{wait}}\}.$$

The mutual informations with Gaussian and BPSK inputs are

$$I(X_{\text{G}}; Y, H = h) = \log_2\left(1 + a^2 \text{SNR}\right), \tag{2.40}$$

$$I(X_{\text{BPSK}}; Y, A = a) = J\left(\frac{2a^2}{\sigma_b^2}\right), \tag{2.41}$$

where the signal to noise ratio is defined by $\text{SNR}=\frac{P}{\sigma_b^2}$ and the function $J(\cdot)$ is defined by eq. (2.15).

The delay probability curves of a Rayleigh fading channel are represented in Fig. 2.28. For a $p_{\text{wait}} = 10^{-2}$ and SNR=15dB the curve is approximately at $\Delta^*(p_{\text{wait}}) = 2$. This means that, when the receiver has collected twice the amount of information bits, or equivalently when the current coding rate equals 1/2, the probability that the decoder has to collect more data to be able to decode is equal to 10^{-2}.

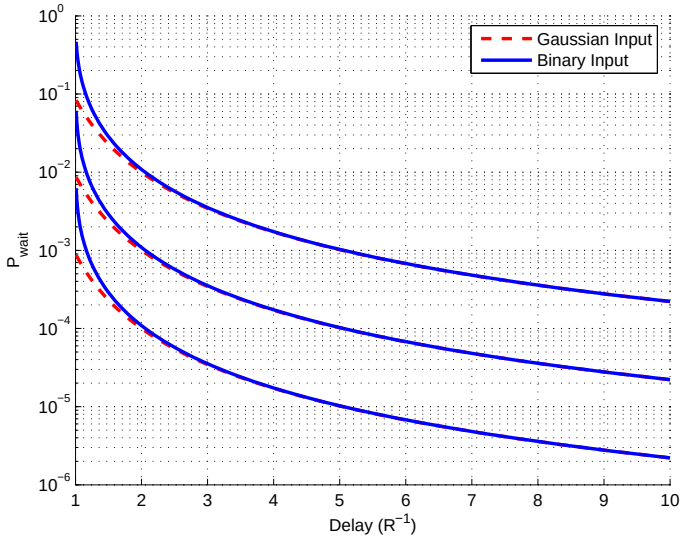


Figure 2.28 – Theoretical limits of rateless schemes over Rayleigh fading quasi-static channels for three values of SNR (10dB, 15dB, 20dB)

2.7.4 Optimization of Raptor codes for quasi-static fading channels

Raptor codes are known *not* to be universal [ES06] on general binary input memoryless channels other than the BEC, *i.e.* they cannot approach arbitrarily close the capacity *independently* of the channel statistic. To address this issue "Generalized Raptor codes" [PNF06] constructions allow the output degree distribution to be change as the output symbols are sent over the channel. The authors show with information theoretic arguments that their construction has the potential to approach the capacity in a rate compatible way, nevertheless no explicit construction or optimization methods is proposed.

However, we will see that "classical" Raptor codes are good candidates to implement rateless coding, since these can operate close to the instantaneous capacity over a wide range of channel gains.

A Raptor code is entirely characterized by its output degree distribution. The optimization of an output degree distribution arises from the characterization of the decoder. Our concern in this section is to provide an optimal approach for the design of such

codes for quasi-static fading channels. We define a cost function for the optimization of the output degree distribution and derive an optimization method that can be stated as a linear problem.

2.7.4.1 Cost function

We now assume that a system designer wants to guarantee a maximal delay for $(1 - p_{\text{wait}})\%$ of users, which means that any information delay $\Delta \leq \Delta^*(p_{\text{wait}})$ must be tolerated (see Section 2.7.3). Thus, good codes must operate close to the instantaneous capacity of the set of channel realizations $\Lambda(p_{\text{wait}})$, which corresponds to the desired range of information delays. An output degree distribution satisfying this can be found as the solution to the following expression

$$\underline{\Omega}_{opt}(p_{\text{wait}}) = \arg \inf_{\underline{\Omega} \in \Xi} \sup_{a \in \Lambda(p_{\text{wait}})} \frac{R^{-1}(a, \underline{\Omega}) - C^{-1}(a)}{C^{-1}(a)} \tag{2.42}$$

where Ξ defines the ensemble of distributions such that the Belief Propagation (BP) decoder can converge on the desired range of capacities. In other words, we search in the class of output degree distributions that converge on the corresponding range of capacities and minimize the *rate overhead*, i.e. the gap between the coding rate R and the instantaneous capacity.

2.7.4.2 Optimization problem statement

Let us now formulate the optimization problem for a quasi-static fading channel. In this scenario, the encoder sees a channel with random capacity. Hence, Raptor codes that perform well on such channel must perform *simultaneously* well over different BIAWGN channels corresponding to a wide range of capacities. In order to guarantee the convergence of the decoder for such *range* of channel capacities $[C_0(p_{\text{wait}}), 1)$, we show how to write the IC evolution equations. In section 2.2, we developed a method that enables to solve the constraint for one given channel capacity. We now rely on such results to constrain the optimization problem *simultaneously* for different channel capacities.

The equations of IC evolution in section 2.2 are only valid for a given value of the average input symbol degree α, in the distribution $\iota(x)$. The α parameter is fixed in order to deal with a linear problem and it is chosen to minimize the overhead. Moreover, this parameter represents the average degree of an input symbol and, as the output symbols are received the input symbols become more connected in the decoding Tanner graph. More precisely, for a given distribution $\Omega(x)$, the quantity α depends on the operating rate $R = \Omega'(1)/\alpha$, which can also be written as $\alpha = \alpha_0 R^{-1}$ where α_0 is the average degree when $R = 1$ (*i.e.* when the number of received output symbols equals the number of input symbols). If we want to guarantee convergence for different

capacities, we have to account for the fact that the average input degree α varies with the instantaneous operating rate. To this end, we make the assumption that the Raptor code operates close to the capacity, i.e $R \approx C$. Consequently, given a capacity value C, the parameter α writes as $\alpha = \alpha_0 C^{-1}$, where α_0 becomes the design parameter. By discretizing the range of capacities $[C_0, 1)$ into $\{C_i\}$ the optimization problem is finally written as follows:

$$\omega_{opt}(x) \quad = \quad \arg\min_{\omega(x)} \sum_j \frac{\omega_j}{j}, \tag{2.43}$$

subject to the constraints:

$[\mathrm{C}_1]$ $\sum_i \omega_i = 1$,

$[\mathrm{C}_2]$ $F(x, C_i) > x \quad \forall x \in [0; x_0 - \delta_C] \quad$ for C_i,

$[\mathrm{C}_3]$ $F(0, C_i) > \varepsilon \quad$ for some $\varepsilon > 0$,

$[\mathrm{C}_4]$ $F'(0, C_i) > 1$.

Similarly to the optimization for the BIAWGN channel, it appears that there is an optimal value for the parameter α_0, *i.e.* there is a value such that the cost function is minimized.

2.7.5 Simulation results

We now provide numerical results to analyze the performance of Raptor codes over quasi-static Rayleigh fading channels under delay constraints. In particular, we optimized an output degree distribution for a quasi-static fading channel with SNR=12dB and outage probability $p_{\text{wait}} = 10^{-4}$. Fig. 2.29 shows that this leads to a maximum delay $\Delta = 10$, or equivalently, a range of capacities equal to $[0.1, 1)$. We optimized an output degree distribution for this range and ran simulations. The overhead, defined as the gap to the capacity (right-hand side of equation (2.42)) is computed with IC evolution. The simulations in Fig. 2.29 show that, for the desired range of capacities, the optimized distribution operates within 10% of the instantaneous capacity.

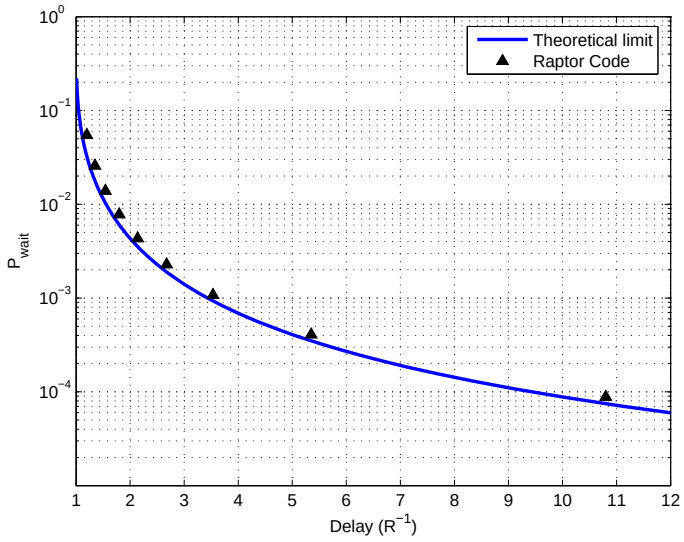


Figure 2.29 – Asymptotic performance, in terms of p_{wait} versus delay, of a Raptor code optimized for a quasi-static Rayleigh fading channel (SNR=12dB) with perfect CSIR. Theoretical limits are also reported.

2.8 Raptor codes with higher order modulations

We first present some simulation results with a 16-QAM modulation for two channels studied in the previous sections, namely the AWGN channel, and the uncorrelated Rayleigh fading channel. We do not distinguish between well protected and bad protected bits, and consider here the average channel seen by the bits at the output of the demapper. In that case, turbo demapping would bring no performance improvement with a Gray mapping, and therefore the distributions are optimized for BPSK inputs.

Then, we also study the case where the receiver has only access to a noisy estimate of the channel. A characterization of channel estimation errors is used in an optimal manner to derive a consistent measure of information for decoding with imperfect CSI. Then we show that the optimal *consistent measure of information* (using channel estimation accuracy) for decoding with imperfect channel estimation is given by the log-likelihood ratio (LLR) of the received bit via a composite (more noisy) channel. Finally, we formalize the problem of decoding with imperfect channel estimation by deriving the optimal LLRs.

2.8.1 Simulation results with 16-QAM input and AWGN channel

In this section, we consider an AWGN channel with output symbols mapped to 16-QAM symbols. The channel is of capacity $C = 0.5$ ($E_s/N_0 = 3.2503$). The purpose of this section is to evaluate how robust the distributions are when considering higher order modulations. The simulation results in Fig. 2.30 to 2.33 present the performance of Raptor codes of information length $K = 1024, 2048, 4096$ and 8192 input bits respectively. The output degree distributions are the same as in section 2.4.

Even though the output degree distributions were not designed for high order modulations, the simulation results show that the performance – in terms of FER vs. overhead – over an AWGN with 16-QAM input alphabet is very similar to the performance over the BIAWGN channel (Fig. 2.14 to 2.17) In particular, the use of lower rate precodes is an efficient technique to design Raptor codes that perform well in the error floor region, with no significant loss in the waterfall region.

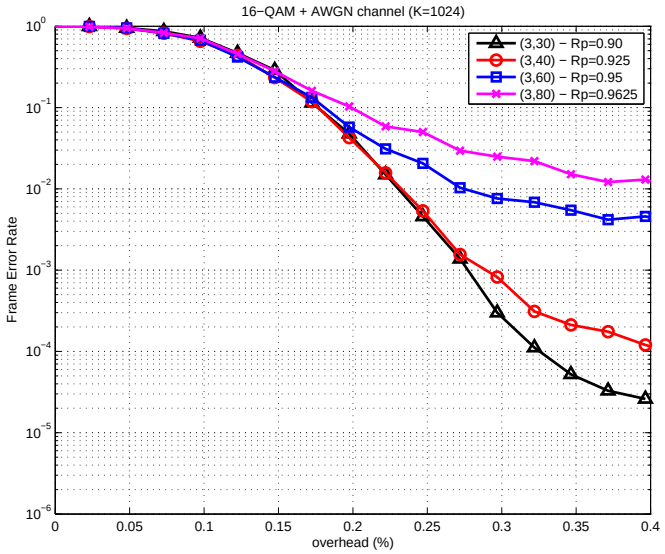


Figure 2.30 – Performance, in terms of FER versus overhead of Raptor codes of size $K = 1024$ over an AWGN channel with 16-QAM input. The channel is of capacity $C = 0.5$

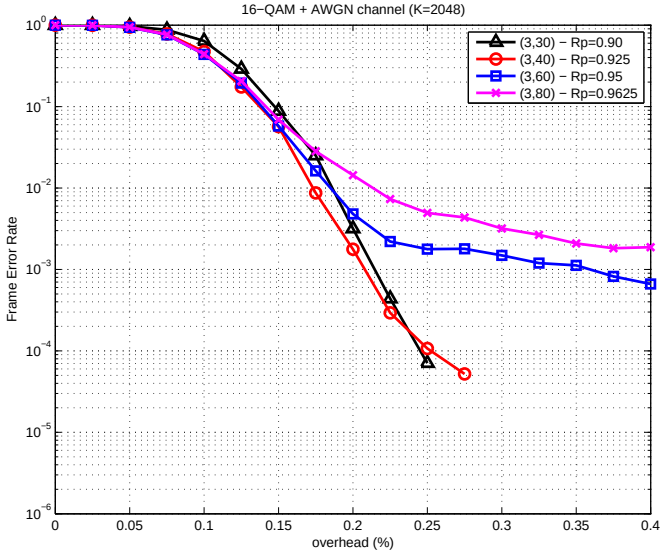


Figure 2.31 – Performance, in terms of FER versus overhead of Raptor codes of size $K = 2048$ over an AWGN channel with 16-QAM input. The channel is of capacity $C = 0.5$

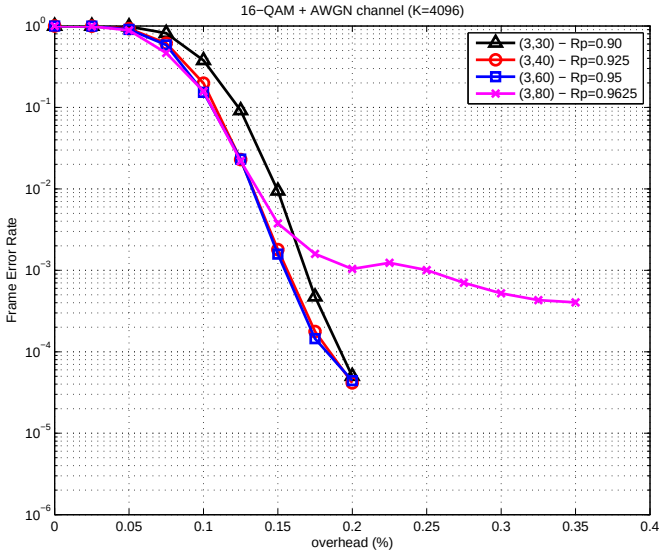


Figure 2.32 – Performance, in terms of FER versus overhead of Raptor codes of size $K = 4096$ over an AWGN channel with 16-QAM input. The channel is of capacity $C = 0.5$

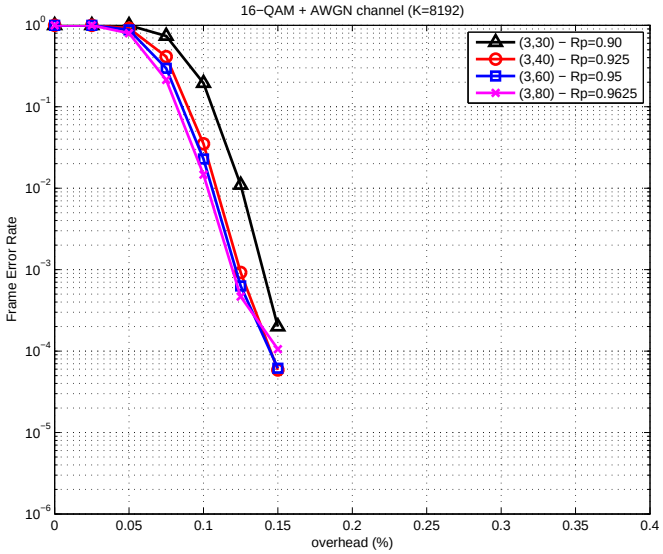


Figure 2.33 – Performance, in terms of FER versus overhead of Raptor codes of size $K = 8192$ over an AWGN channel with 16-QAM input. The channel is of capacity $C = 0.5$

2.8.2 Simulation results with 16-QAM input and uncorrelated Rayleigh fading channel

Simulation results in section 2.6.3 showed that for BPSK inputs, Raptor codes have the same performance over AWGN channel and over an uncorrelated Rayleigh fading channel. We now consider an uncorrelated Rayleigh fading channel with 16-QAM inputs. The output degree distributions are the same as in section 2.4. The simulation in Fig. 2.34 to 2.37 show that with channel inputs in a 16-QAM constellation, Raptor codes have the same performance on an uncorrelated Rayleigh fading channel than with an AWGN channel (previous section, Fig. 2.30 to 2.33)

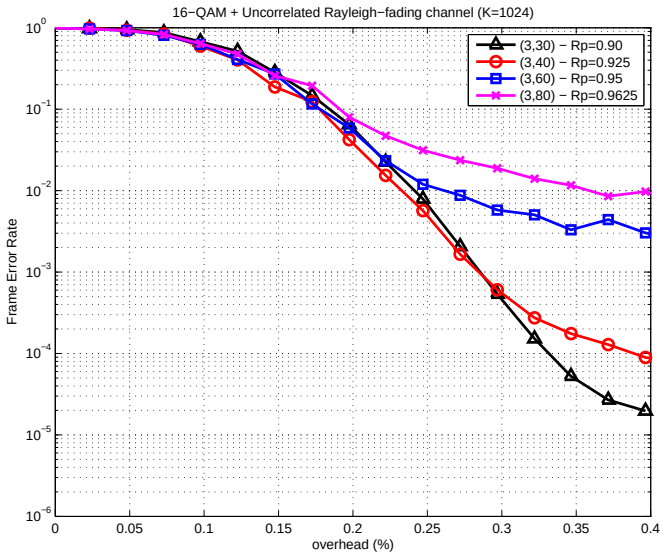


Figure 2.34 – Performance, in terms of FER versus overhead of Raptor codes of size $K = 1024$, over an uncorrelated Rayleigh fading channel with 16-QAM input. The channel is of capacity $C = 0.5$ ($E_s/N_0 = 4.937$)

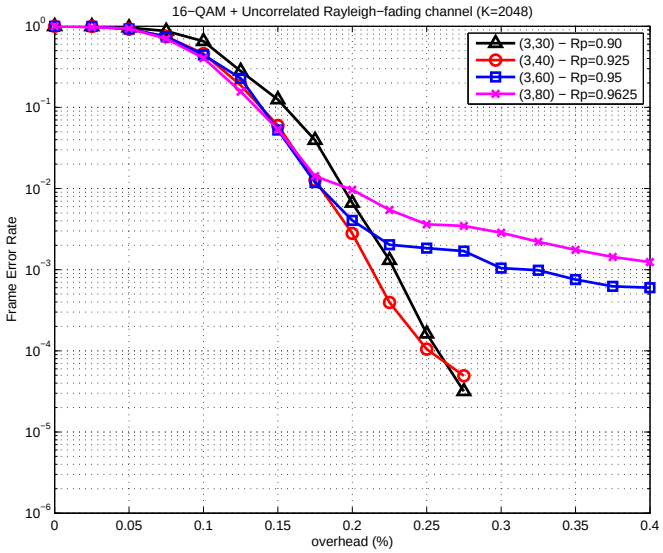


Figure 2.35 – Performance, in terms of FER versus overhead of Raptor codes of size $K = 2048$, over an uncorrelated Rayleigh fading channel with 16-QAM input. The channel is of capacity $C = 0.5$ ($E_s/N_0 = 4.937$)

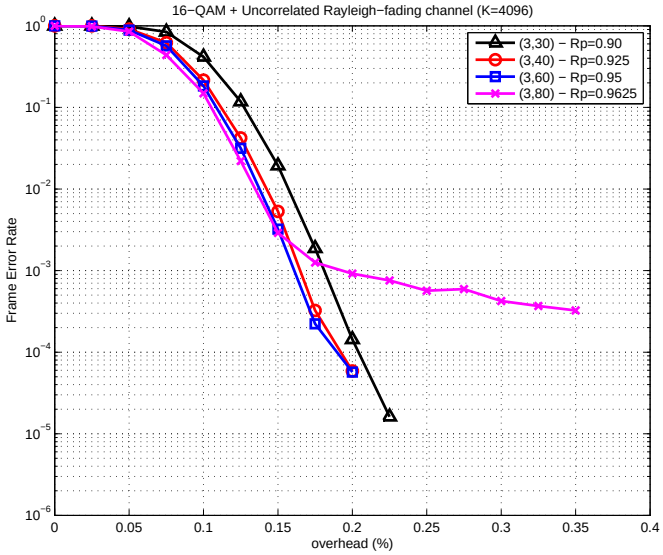


Figure 2.36 – Performance, in terms of FER versus overhead of Raptor codes of size $K = 4096$, over an uncorrelated Rayleigh fading channel with 16-QAM input. The channel is of capacity $C = 0.5$ ($E_s/N_0 = 4.937$)

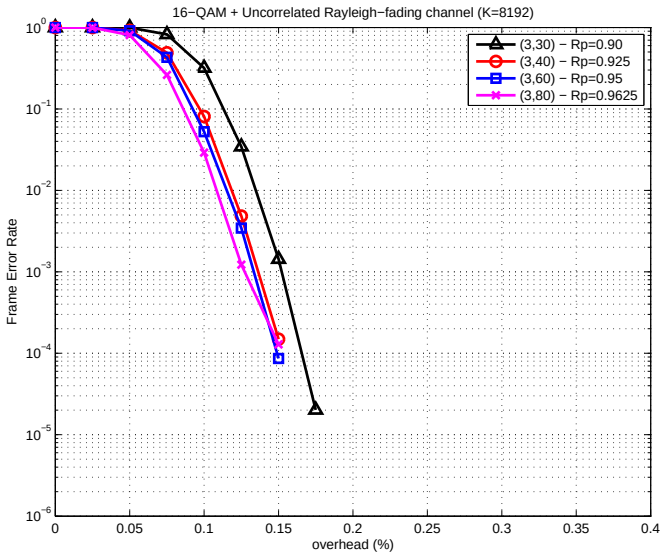


Figure 2.37 – Performance, in terms of FER versus overhead of Raptor codes of size $K = 8192$, over an uncorrelated Rayleigh fading channel with 16-QAM input. The channel is of capacity $C = 0.5$ ($E_s/N_0 = 4.937$)

2.8.3 Decoding Raptor codes with imperfect CSIR

In the previous sections, we assumed that perfect channel knowledge is available at the receiver. However, this assumption is not longer valid in most of practical wireless systems. This motivates us to study the design of optimal Raptor codes with imperfect channel estimation. The main consequence of channel estimation errors is that perfect coherent demodulation is not possible any more, which can severely affect the system performance. Note that since it is impossible to perform a perfect demodulation, it is necessary to use complex input and output alphabets in the channel model.

Let us review models for communications over memoryless channels with complex input and output alphabets. A specific instance of the channel is characterized by the transition probability density $W(y|x,H) = C\mathcal{N}(Hx, \sigma_Z^2)$ with channel state $H \in \mathbb{C}$:

$$Y = HX + N$$

Channel estimation: The transmitter, before sending the data, can teach the channel to the receiver by sending a training sequence of N_T symbols $\mathbf{x}_T = (x_{T,1}, \ldots, x_{T,N_T})^T$. For quasi-static fading, the coherence time of the channel is much longer than the training time. Moreover, we denote the average energy of the training symbols by $P_T = \frac{1}{N} tr\big(\mathbf{x}_T \mathbf{x}_T^\dagger\big)$. This sequence is affected by the channel gain H, allowing the receiver to perform ML estimation of H from the observed signals $\mathbf{y}_T = H\mathbf{x}_T + \mathbf{z}_T$ and $\mathbf{x}_T$. This yields to $\widehat{H} = H + \mathcal{E}$, where $\mathcal{E}$ denotes the estimation error of variance $\sigma_{\mathcal{E}}^2 = \mathrm{SNR}_T^{-1}$ with $\mathrm{SNR}_T = \frac{N_T P_T}{\sigma_Z^2}$. The receiver only knows the estimate $\widehat{H}$ and a characterization of its accuracy in terms of the conditional pdf $P_{H|\widehat{H}}$. This pdf can be obtained by using the likelihood function, the pdf $W(y|x,H)$, and P_H and is given by:

$$P_{H|\widehat{H}} = C\mathcal{N}(\delta\widehat{H}, \delta\sigma_{\mathcal{E}}^2), \quad \text{with } \delta = \sigma_H^2/(\sigma_H^2 + \sigma_{\mathcal{E}}^2). \tag{2.44}$$

In [PSD07] the authors used the estimation accuracy given by (2.44) to derive a ML decoder that minimizes the average error probability over all channel estimation errors. They also show that the mismatched ML decoder is not adapted to imperfect channel estimation. Furthermore, their decoder achieves the capacity of a composite channel.

These ideas will serve as the basis to address the problem of the mismatched LLR in Raptor codes. The mismatched LLR simply consists in replacing the unknown channel gain by its estimate in the standard LLR expression. However, this operation does not lead to a consistent measure of information, namely the LLR densities do not satisfy the symmetry condition (see *e.g.* discussions in [HSM01]). We first review the mismatched LLR to then derive a new LLR adapted to the channel estimation errors that leads to a consistent measure of information, which is an important assumption for characterizing the BP decoder under Gaussian approximation with IC evolution.

2.8.3.1 Mismatched LLRs

For simplicity, the computations are given for QPSK modulation only. The mismatched log-likelihood ratios, which consists of replacing H by its estimate $\widehat{H}$, are computed as follows:

$$L_M(y,\hat{h}) = \log \frac{W(Y_i = y|X_i = 1, H = \hat{h})}{W(Y_i = y|X_i = -1, H = \hat{h})} = \frac{4\hat{h}^\dagger y}{\sigma_Z^2}. \tag{2.45}$$

This LLR is complex, and the LLR corresponding the to inphase (I) and quadrature (Q) branches are respectively given by real(L_M) and imag(L_M). An LLR is said a consistent measure of information if its conditional density is *symmetric* [RU01]. For a QPSK modulation and a quasi-static fading channel, the symmetry condition implies that the variance $\mathrm{Var}\big(L_M(y,\hat{h})|X=1,\widehat{H}=\hat{h}\big)$ is equal to $4\,\mathbb{E}_{Y|X\widehat{H}}\big\{L_M(y,\hat{h})|X=1,\widehat{H}=\hat{h}\big\}$.

It is not difficult to compute these quantities:

$$\begin{aligned}
&\mathbb{E}_{Y|X\widehat{H}}\big\{L_M(y,\hat{h})|X=1,\widehat{H}=\hat{h}\big\} = \frac{4\,\delta|\hat{h}|^2}{\sigma_Z^2},\\
&\mathrm{Var}\big(L_M(y,\hat{h})|X=1,\widehat{H}=\hat{h}\big) = \frac{4^2\,|\hat{h}|^2(\sigma_Z^2+\delta\sigma_{\mathcal{E}}^2)}{(\sigma_Z^2)^2},
\end{aligned} \tag{2.46}$$

which shows that in presence of imperfect channel estimation, the mismatched LLR does not lead to a consistent measure of information.

2.8.3.2 LLRs using channel estimation accuracy

We now adapt the LLRs to the channel estimation errors. To this end, we compute the log-likelihood ratio using the composite channel obtained by averaging the original channel $W(y|x,H)$, which depends on the unknown gain H, over the *a posterior* pdf of H given $\widehat{H}$. After some algebra [PSD07], we obtain:

$$\widetilde{W}(y|x,\widehat{H}) = C\mathcal{N}\big(\delta\widehat{H}x, \sigma_Z^2 + \delta\sigma_{\mathcal{E}}^2|x|^2\big), \tag{2.47}$$

where $\delta = \frac{1}{1+\sigma_\epsilon^2}$. The averaged channel contains some additional noise related to the estimation errors. Actually, we can derive the LLR expression corresponding to the received bits via the composite channel:

$$\begin{aligned}
L_{\mathrm{opt}}(Y_i = y,\hat{h}) &= \log \frac{\widetilde{W}(Y_i = y|X_i = 1,\widehat{H}=\hat{h})}{\widetilde{W}(Y_i = y|X_i = -1,\widehat{H}=\hat{h})},\\
&= \frac{4\,\delta\hat{h}^\dagger y}{\sigma_Z^2+\delta\sigma_{\mathcal{E}}^2}.
\end{aligned} \tag{2.48}$$

Hence, the variance and the mean of (2.48) satisfy the symmetry condition:

$$\begin{aligned}
\mathrm{Var}\big(L_{\mathrm{opt}}(y,\hat{h})|X=1,\widehat{H}=\hat{h}\big) &= \frac{4^2\,\delta^2|\hat{h}|^2}{\sigma_Z^2+\delta\sigma_{\mathcal{E}}^2}\\
&= 4\,\mathbb{E}_{Y|X\widehat{H}}\big\{L_{\mathrm{opt}}(y,\hat{h})|X=1,\widehat{H}=\hat{h}\big\}.
\end{aligned} \tag{2.49}$$

which implies that (2.48) is Gaussian and defines a consistent measure of information.

2.8.3.3 Simulation results

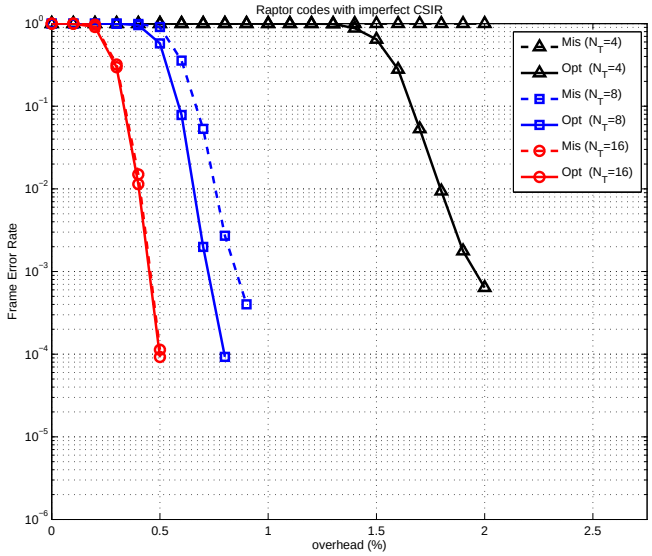


Figure 2.38 – Performance of Raptor codes with imperfect channel estimation at the receiver on uncorrelated Rayleigh fading channel with 16-QAM modulation. Channel estimation is performed with $N_T = 4, 8, 16$ pilots. Codes decoded with a mismatched decoder are labeled 'Mis' (dashed lines), and codes decoded with the proposed metric are labeled 'Opt' (solid lines).

In this setting, we consider that the fast fading model is the limit of a fully interleaved bloc fading model, and that pilot based channel estimation is performed for each bloc. Fig. 2.38 presents the performance of Raptor codes with information length $K = 4096$, with the following parameters for the channel estimation: $N_T = 4, 8$ and 16 pilots.

For the case of $N_T = 4$ pilots, the mismatched decoder is unable to perform decoding, whereas the decoder that used the proposed metric converges. The convergence occurs for very large overhead, nevertheless this observation has to be balanced by the fact that overheads are computed relatively to the channel capacity with perfect CSIR. Therefore the channel capacity in presence of imperfect channel estimation is lower, and in that case, the relevant capacity is the *estimation induced capacity* presented in [Pia07, PMD]. For the case of $N_T = 8$ pilots, the decoder that used the proposed metric shows a performance improvement of 1 decade. For the case of $N_T = 16$ pilots, both decoders exhibit similar performance (the mismatched decoder is slightly degraded) since with $N_T = 16$ pilots the channel estimation is quite accurate.

Fig. 2.39 shows the performance of Raptor codes with information length $K = 8192$, and 16-QAM input alphabet, over a quasi-static Rayleigh fading channel, in the scenarios of: (i) imperfect CSIR by using the mismatched LLRs (expression (2.45)) and (ii) imperfect CSIR by using the LLRs with channel estimation accuracy (expression (2.48)). These results indicate that mismatched LLRs are sub-optimal for short training sequences, and confirmed the adequacy of the improved LLRs. This performance improvement was obtained without introducing any additional complexity.

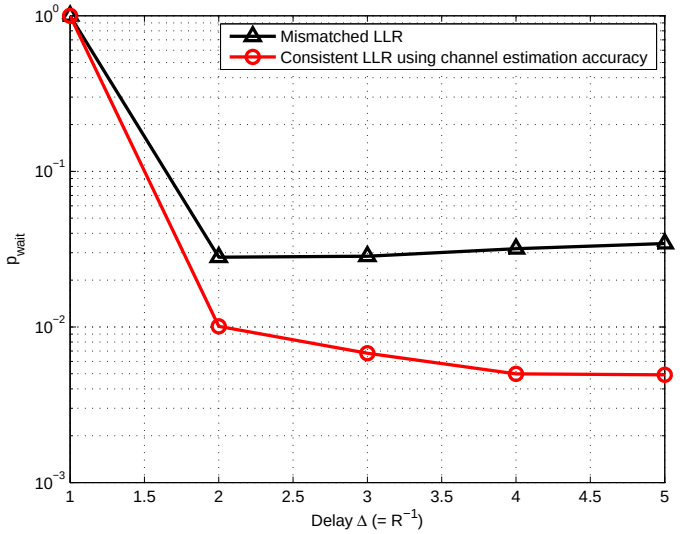


Figure 2.39 – Finite length performance of Raptor codes with imperfect channel estimation at the receiver ($N_T = 1$ training symbol and $P_T = P$).

2.9 Summary

In this chapter, we first presented an analytical asymptotic analysis of Raptor codes over a BIAWGN channel. Based on the analysis, we derived an optimization method for the design of efficient output degree distributions. The optimization problem is linear with respect to the coefficients of the output degree distribution and therefore it can efficiently be solved with linear programming. We analysed the main design parameters by showing their influence in the optimization process. We showed – with threshold computations and FER vs. overhead simulations – that even though Raptor codes are not universal on other channels than the BEC, Raptor code optimized for a given channel capacity also perform well on a wide range of channel capacities when joint decoding is considered.

We addressed the issue of finite length design with a rate splitting strategy, and showed that the use of lower rate precodes ($R_p \simeq 0.9$) is a valid strategy for designing Raptor codes that perform well at small to moderate lengths. The rate splitting strategy is also valid on the BEC. An asymptotic analysis of the joint decoder and the rate splitting study for the BEC are presented in appendix B

Then, we investigated the extension of the analysis to the uncorrelated Rayleigh fading channel with perfect CSIR. We showed that with IC evolution, the quantity of interest in the channel model is the ergodic capacity. Therefore the analysis is not specific to the BIAWGN, rather the IC evolution analysis is correct for a channel of a given capacity. Consequently, the optimization method derived for the BIAWGN can be used for the uncorrelated Rayleigh fading channel, and we showed with threshold computations and finite length simulations that distributions optimized for a (BIAWGN) channel of capacity $C = 0.5$ perform well on an uncorrelated Rayleigh fading channel of capacity $C = 0.5$.

The optimization method for the BIAWGN channel served as a basis for the optimization of Raptor codes for quasi-static fading channels, which consists in optimizing a degree distribution that operates close to the channel capacity *simultaneously* for different channel capacities. We showed that it is possible to constrain the optimization problem such that convergence of the decoder is enforced *simultaneously* for different channel capacities.

Finally, we investigated the behavior of various distributions over higher order modulations. Even though the degree distributions were optimized for BPSK modulation, we showed with simulation results that they also perform well with higher order modulations, for both the AWGN channel and the uncorrelated Rayleigh fading channel. We showed that in presence of imperfect CSIR, it is possible to improve the performance with no additional complexity, by using an appropriate metric for the computation of the LLR at the output of the channel.

3

Low-Density Parity-Check construction algorithms

At small to moderate block lengths, randomly constructed LDPC codes perform rather poorly because of shorts cycles in the associated Tanner graph. The performance of a code is closely related to the girth of the graph, which is defined as the size of the shortest cycle, and therefore, it is necessary to construct graphs with the largest girth possible. In this chapter, we present some LDPC construction algorithms – both pseudorandom constructions, and structured constructions – that can be used to design efficient LDPC (pre)codes for small to moderate block lengths.

In the previous chapter, we presented some finite length simulations of Raptor codes. The Low-Density Parity-Check (LDPC) were used as precodes were constructed with an algorithm, namely the RandPEG, that we present in this chapter.

3.1 Introduction

LDPC codes, originally introduced by Gallager [Gal63, Gal62], are a class of linear block codes described by a sparse parity-check matrix H and can be graphically represented with a bipartite graph, also called Tanner graph: each line of the matrix is associated to a constraint node, and each column of the matrix is associated to a variable node; a variable node i and a check node j are connected by an edge in the Tanner graph iff $H_{i,j} = 1$, *i.e* iff the is a '1' in the corresponding column and line of the parity-check matrix.

Since their rediscovery by Tanner [Tan81] and by MacKay [Mac99] LDPC codes have attracted a lot of attention since they are equipped with an iterative decoding algorithm, namely the Belief Propagation (BP) decoder, that is asymptotically optimal: in the limit of infinite codeword length, the BP decoder gives a maximum *a posteriori* (MAP) estimation of the transmitted codeword. Moreover, LDPC codes can be optimized to operate very close the the channel capacity [LMSS01b, RSU01]. We assume here that the irregularity profile is given and focus on the construction of finite length codes.

Indeed, optimized codes approach capacity only in the asymptotic regime, *i.e.* only in the limit of infinite block lengths. In that regime, the concentration theorem [RU01] ensures that under iterative decoding, the performance of randomly constructed LDPC codes is very close to the asymptotic behavior of the code ensemble. However, at finite length, the local tree assumption[1] is not valid anymore and consequently BP is a suboptimal decoding algorithm. Indeed, the cycles in the Tanner graph which introduce *(i)* correlation between messages (which give rise to a loss in the waterfall region) *(ii)* local structures such as stopping sets [DPT$^+$02] or trapping sets (which give rise to decoding failures and introduce an error floor phenomenon [DPT$^+$02] [Ric03]). More complicated structures induce "pseudo codewords" or "near codewords" [MP03], which are defined as fixed points of the iterative decoder that are *not* codewords.

Therefore, at small to moderate block lengths, random constructions perform rather poorly because of shorts cycles in the associated Tanner graph, especially for irregular codes, because the fraction of variable nodes with a large degree induce many short cycles. At finite length, the performance of a code is closely related to the girth of the graph, which is defined as the size of the shortest cycle. The motivation behind having high girth LDPC codes is twofold. Firstly, the BP decoder tends to MAP decoder and thus behaves better in the waterfall region. Secondly, an LDPC code with few short cycles has less chances to exhibit decoding failures caused by pseudo-codewords than an LDPC code with many short cycles. Therefore, a high girth tends to improve the performance in the error floor region. Moreover, the importance of constructing graphs with high girth is particularly important for non-binary (NB) LDPC codes. The BP decoder and its simplified versions have shown to achieve very good performance on non-binary GF(q)-LDPC codes with $d_v = 2$ and with high order Galois fields [HE03, DF05]. For these ultra-sparse codes, it is crucial to focus on the girth properties of the underlying Tanner graph [PFD07].

Large girths are obtained by suitably constructing the underlying Tanner graph, and there are two main approaches for the construction of Tanner graphs: pseudorandom constructions and algebraic constructions. Let us now briefly review the two approaches.

[1] The local tree assumption states that the girth of the graph is large enough so that the local neighborhood of any variable node is a tree (there are no repeated nodes in the subgraph)

Pseudorandom constructions A pseudorandom construction based on a progressive edge-growth (PEG) of the graph was proposed in [HEA05], which results in LDPC codes that have higher girths compared to pre-existing random LDPC code construction techniques. For the construction of irregular LDPC codes, an Extrinsic Message Degree (EMD) metric and its approximation metric called Approximate Cycle EMD (ACE) metric was introduced in [TJVW04a, TJVW04b]. The ACE graph construction consists in putting the high degree nodes in the short cycles, thus eliminating small cycles with low EMD, such that much extrinsic information is propagated through these problematic cycles. The PEG algorithm has been extended to include the ACE metric, which improves the performance of irregular LDPC codes in the error floor region [XB04].

Structured constructions First we mention algebraic constructions include Euclidean Geometry (EG) codes and Projective Geometry (PG) codes [KLF01]. The parity-check matrices of EG (resp. PG) codes are defined with the points and lines of a euclidean (resp. projective) geometry over finite fields. A full discussion on EG/PG codes is far beyond the scope of this chapter, but one interesting property of EG/PG codes is that they can be put in a cyclic or quasi-cyclic (QC) form, which is a structured construction of interest. The Tanner graph of such codes is of girth 6 by construction. Some aspects of the construction of QC LDPC codes will be discussed later in this chapter.

3.2 Randomized Progressive Edge-Growth

In this section, we propose some improvements in the PEG algorithm which greatly improve the girth properties of the resulting graphs: given a graph size, they increase the girth g achievable by the algorithm, and when the girth cannot be increased, our modified algorithm minimizes the number of cycles of length g. As a main illustration, we focus on regular column-weight two graphs ($d_v = 2$), although our algorithm can be applied to any graph connectivity. The class of $d_v = 2$ graphs is often used for non-binary low density parity check codes that can be seen as monopartite graphs: for a given target girth g_t, this new instance of the PEG algorithm, that we call RandPEG, allows to construct cages, *i.e.* graphs with the minimal size such that a graph of girth g_t exists, which is the best result one might hope for.

3.2.1 Notations and definitions

A bipartite graph is denoted as (V, E) where V (resp. E) is the set of the vertices (resp. edges). $V = V_c \cup V_s$ where V_c is the set of check nodes and V_s the set of symbol nodes. Let $N = |V_s|$ denote the total number of symbol nodes, which we will refer to as the size of the graph. When the graph is the Tanner graph of an LDPC code, N is

the codeword length. For a given graph setting, namely a 3-tuple (d_v, d_c, g), we denote by $N_g^{(d_v,d_c)}$ the lower bound on N such that a (d_v, d_c) regular graph of girth g exists. This lower bound can be easily computed by using the results of [HEA05, lemma 3], and is known *not* to be tight when $d_v = 2$, for $g \geq 18$ [Big88].

The original PEG algorithm [HEA05] is a procedure for constructing a bipartite graph in an edge by edge manner, where the selection of each new edge aims at minimizing the impact on the girth: at each step the local girth is maximized. Let $\mathcal{N}_{s_j}^l$ denote the set of all check nodes reached by a tree spanned from symbol node s_j within depth l, and $\bar{\mathcal{N}}_{s_j}^l$ denote the complementary set in V_c. We use the same definition as in [HEA05] for the depth. At a given stage of the construction, only a subset of the check nodes have reached a connectivity of d_c, and we call *candidates* the check nodes in $\bar{\mathcal{N}}_{s_j}^l$ whose incident edges have not been all assigned. When a particular check node is *selected* among the candidates, an edge is added in the graph between the node s_j and that check node.

For each node s_j, the first edge is chosen randomly, and the other edges are chosen in the set $\bar{\mathcal{N}}_{s_j}^l$, where l is such that $\bar{\mathcal{N}}_{s_j}^l \neq \emptyset$ and $\bar{\mathcal{N}}_{s_j}^{l+1} = \emptyset$, *i.e.* among the nodes that are at the largest depth from the symbol node s_j. This maximizes the length of the cycles created through this new edge. When multiple choices are possible, the algorithm selects the candidate that has the smallest degree under the current setting. Selecting a candidate that is at a depth l from node s_j creates a cycle of length $2l + 2$.

Even though the original PEG algorithm produces only *almost* regular graphs, the construction of *strictly* regular graphs can be easily enforced by discarding all candidates where all the edges have already been assigned. In the sequel, we only consider the construction of regular (d_v, d_c) graphs, in order to compare to the known bounds for regular graphs. We emphasize the fact that this limitation concerns only our study, *not* the RandPEG algorithm itself.

3.2.2 The RandPEG Algorithm

In this section, we describe our contributions in details. There are basically two differences between the original PEG algorithm and the RandPEG algorithm that we propose: firstly, the way we build and use the spanning tree is different, and secondly, we introduce an objective function that we use for the edge selection. The RandPEG algorithm is based on a randomization approach: given a target girth g_t, we consider, at each stage of the construction, the maximum number of possibilities when adding an edge in a graph, and we use an objective function to discriminate among the numerous edge candidates. Our goal is to actually reach a given target girth g_t of the bipartite graph, when *all* the edges of the graph have been assigned. Therefore, if at some point of the construction there is no possibility to add an edge without creating

a short[2] cycle, then we consider that the algorithm *fails*. In that case, all the edges in the graph are discarded and the algorithm starts from scratch. Similarly to Monte Carlo approaches, the algorithm runs many times and stores the best graph.

3.2.2.1 Truncated spanning tree

Instead of spanning to the maximal possible depth, we span the tree only up to a maximal depth l_{max}. This technique, which defines the *nongreedy* version of the algorithm [HEA05], is suggested for the construction of long codes where it would be computationally expensive to build the whole tree. Here, we argue that this is not only a computational or speed-up enhancement of the algorithm, but that this technique *should* be used when one wants to construct a graph that matches the lower bound $N_g^{(d_v,d_c)}$. We justify our argument with the following three points.

Diameter argument First, we give a justification on how deep the construction tree should be spanned, based on a graph argument: for a given value of the target girth g_t, if the graph has minimum size $N = N_{g_t}^{(d_v,d_c)}$ then the diameter of the graph equals $d = g_t/2$ [TSF01]. Therefore in that case, the tree *must* be spanned up to a maximal depth $l_{max} = (g_t - 2)/2$, so that the diameter is ensured to equal $d = g_t/2$. Indeed, if at some point the algorithm selects a node in $\bar{\mathcal{N}}_{s_j}^{l}$ with $l > g_t$, then the condition that diameter of the graph equals $g_t/2$ cannot hold, and the construction will fail.

The spanning of the tree at a given depth $l = (g_t - 2)/2$ gives a set of candidates for which we ensure that no cycle smaller than the target girth g_t can be created if such a candidate is selected.

The randomization approach We recall that our goal is to reach a given target girth g_t, when *all* the edges of the graph have been assigned. By spanning the tree less deeply, the number of candidates at each step of the algorithm becomes much larger, and each edge is selected among a very large number of candidates. Thus, the algorithm is based on a certain amount of randomness in the construction: if at some point the construction fails, then all the edges are discarded and the procedure restarts from scratch. This justifies the name of "Randomized PEG", and ensures that a wide variety of solutions are explored.

Reduced probability of construction failure When spanning the tree to its maximal depth, the first cycles that are created by the algorithm are locally optimal in the sense that they are of the largest possible size. However, as the procedure progresses, the construction problem becomes too constrained and eventually fails if the target girth is relatively high compared to the graph parameters. Our extensive tests show that by

[2] by short cycle, we mean cycles that are shorter than the target girth.

spanning the tree at a lower depth, we create smaller cycles at the beginning of the procedure and thus the choice of the edge is *not* locally optimal, but nevertheless the probability that the algorithm actually terminates is much higher.

3.2.2.2 The objective function

We consider in this section the general case where $N \geq N_g^{(d_v,d_c)}$, *i.e.* when the graph size N is large enough such that a (d_v, d_c) graph of girth g may exist. The set of candidates can be potentially very large, especially at the beginning of the graph construction, and it becomes possible (and necessary) to discriminate among the multiple candidates.

We describe here the objective function that we used, which minimizes the number of created cycles. We would like to point out that other objective functions could be used complementarily: the minimization of other topological structures such as the number of created stopping sets, trapping sets *etc.* or the minimization of an ACE metric, as done in [XB04] for the construction of irregular graphs.

When the construction tree is spanned up to a maximal depth l_{max}, the objective function restricts the set of candidates $\bar{\mathcal{N}}_{s_j}^{l_{max}}$, as follows:

1- If there are candidates at depth l_{max}, then discard all the candidates that are not exactly at the depth l_{max}. By doing so, we only create cycles of size *exactly* l_{max}, and ensure that the diameter argument is fulfilled

2- For each candidate c_j, compute $nbCycles_j$, the number of cycles that would be created if c_j is selected. Discard all candidates that would create more than $min_j(nbCycles_j)$.

3- Compute ${d_c}^{min}$, the lowest degree of all remaining candidates. Discard all candidates with current degree $d_c > {d_c}^{min}$

At this point, the algorithm randomly samples among the remaining candidates.

3.2.2.3 Refinement for spanning the tree

For a given target girth g_t, the diameter argument does not hold anymore for lengths N such that $N_{g_t}^{(d_v,d_c)} < N < N_{g_t+2}^{(d_v,d_c)}$. In that case, the diameter may be larger than $g/2$, and we propose an alternative strategy by introducing a *gap* variable: we span the tree up to a maximal depth $l_{max} = (g_t + gap - 2)/2$.

At the beginning of the construction, cycles of size larger than $g_t + gap$ are created. Each time that it is no longer possible to add any edge, we decrease the value of *gap*, and therefore allow to create smaller cycles. At some point $gap = 0$, then we span the tree only up to a depth $l = (g_t - 2)/2$, and only at this point the algorithm starts creating cycles of size g_t.

This technique, coupled with the objective function described in the previous section,

allows to minimize the multiplicity of the girth, *i.e.* the number of cycles length g_t. A typical value of $gap = 2$ is used for most constructions of regular codes.

Algorithm 3: RandPEG algorithm

```
Data: g_t, N, {(d_{s_j})_{j=1:N}}, w_max, gap
Result: G = (V, E)
w ← 0; /* number of trials for the graph construction */
BEGIN  E ⟵ ∅; gap ← 2
for j ← 1 to N do
    for k ← 1 to d_v do
TRY     SpanTree within maximal depth (g_t + gap − 2)/2
        if N̄_{s_j}^{g_t+gap} = ∅ then
            if gap > 0 then
                gap = gap − 2
                goto TRY
            else
                if w < w_max then
                    w++
                    goto BEGIN
                else
                    Algorithm FAILS
        else
            Apply Objective Function (section 3.2.2.2)
            Randomly select a candidate: E_{s_j}^k ← edge (s_j, c_j)
StoreGraph
```

3.2.3 Performance of the RandPEG algorithm

3.2.3.1 Design of ultra-sparse graphs (d_v=2)

For a given graph setting and a given target girth, there exists a minimal size for the graph such that a graph of girth g_t exists, which is often given in terms of a lower bound. We call *minimal graph* a graph that has a size that achieves the lower bound. In Table 3.4, we report the smallest value of N such that the RandPEG algorithm could construct a regular $(2, d_c)$ graph of girth g, for different values of d_c and g. When this value achieves the lower bound $N_g^{(2,d_c)}$, we indicate so by super-scripting with a star $(*)$, thus indicating that the graph is minimal. Otherwise the value of the lower bound $N_g^{(2,d_c)}$ is super-scripted with parenthesis. Some values are super-scripted with a dag, which means that the RandPEG was initialized with a tree for these constructions. For comparison, the value of N such that the standard PEG algorithm could construct the

corresponding graph is reported in square brackets.

For all values of d_c that we tested up to 50, the RandPEG successfully constructs minimal graphs for target girths $g = 6, 8$. Moreover, for lower values of $d_c = 3, 4$ the algorithm successfully constructs graphs of girth up to 16 that achieve the lower bound. The corresponding graphs can be found on [Dec].

3.2.3.2 Ultra-sparse graphs for NB-LDPC codes

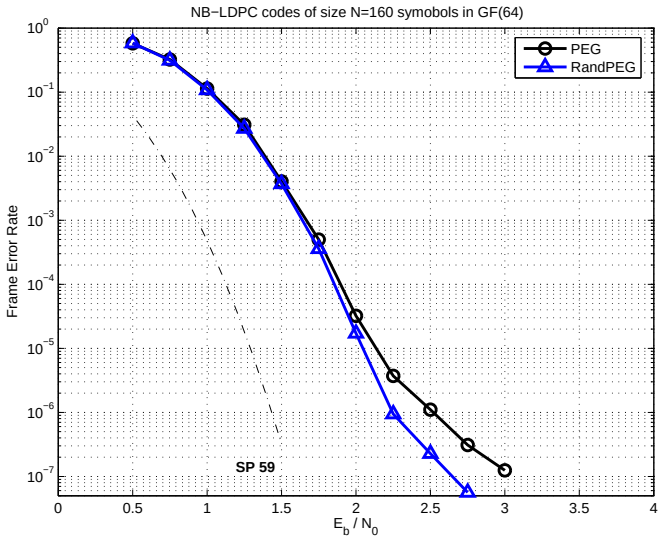


Figure 3.1 – Performance comparison for two non-binary LDPC codes: two codes whose underlying Tanner graphs were constructed respectively with the PEG and RandPEG algorithm are simulated over a BIAWGN channel. The 'SP59' label denotes the Sphere Packing bound of 1959 [Sha59].

We now illustrate the interest of our algorithm for the design of ultra-sparse non-binary LDPC codes, also called codes non-binary cycle Tanner-graph codes [HE03]. We designed two codes of rate one-half, with (2,4) graphs of size $N = 160$. For this graph setting the standard PEG algorithm constructed a graph of girth 12, whereas the RandPEG constructs a minimal graph of girth 16. For both graphs, the non-binary coefficients in GF(64) were optimized according to the method described in [PFD07], and the resulting codes were simulated over a BIAWGN channel with BP decoding. Frame error rates are estimated with 100 frames in error, with a *maximum* of 1000 decoding iterations. The simulation results in Fig. 3.1 show that for ultra-sparse non-binary LDPC codes, a graph with better girth properties performs better in the error floor region, by inducing better spectrum and minimum distance properties [PFD07] The cycle distributions are given in the following tables, and illustrate the importance of the objective function.

	# 12-cycles	# 14-cycles	# 16-cycles	# 18-cycles	# 20-cycles
PEG	58	227	495	1152	3044
RandPEG	0	0	1620	0	5184

Table 3.1 – Cycle distribution for (2,4) column-weight two graphs of size $N = 160$

3.2.3.3 Regular (3,6) codes

The distribution of the stopping/trapping sets determines the behavior of LDPC codes in the error floor region [DPT+02, SCRV06, Ric03], and by minimizing the girth multiplicity, the RandPEG algorithm tends to reduce the number of stopping/trapping sets, and therefore, we expect the resulting LDPC codes to perform better in the error floor region. We used the RandPEG algorithm to construct regular (3,6) LDPC codes, of sizes $N = 504$ and $N = 1008$. The cycle distributions are given in the following tables, and illustrate the importance of the objective function.

	# 4-cycles	# 6-cycles	# 8-cycles
MacKay	169	1312	10052
PEG	0	808	11147
RandPEG	0	449	12027

Table 3.2 – Cycle distribution for (3,6) LDPC codes of size $N = 504$

	# 4-cycles	# 6-cycles	# 8-cycles
MacKay	178	1297	10082
PEG	0	167	10775
RandPEG	0	31	11223

Table 3.3 – Cycle distribution for (3,6) LDPC codes of size $N = 1008$

The resulting codes were simulated on a Binary Symmetric Channel (BSC) and decoded with BP decoding. We compared the PEG and RandPEG constructions to the codes optimized by MacKay [Mac]. The results presented in 3.2 and 3.3 show that indeed the minimization of the girth multiplicity improves the performance of LDPC codes in the error floor region.

3.3 Review of structured LDPC constructions

Quasi-Cyclic codes are well adapted to hardware implementation since they have a simple description that can be stored with minimum memory requirements. Moreover, provided that the parity-check matrix is full rank, the generator matrix of a QC LDPC code can be put in a QC form and in that case QC LDPC codes can be encoded

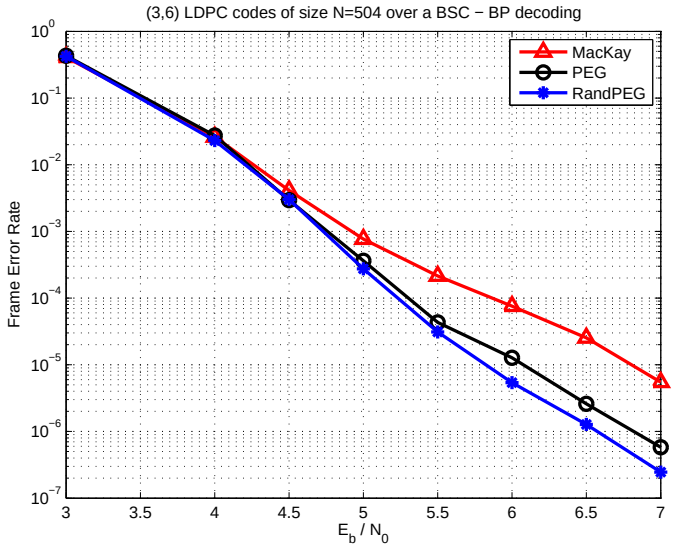


Figure 3.2 – Peg/RandPEG comparison: performance of regular (3,6) LDPC codes of size $N = 504$ simulated on a BSC channel, and decoded with BP decoding

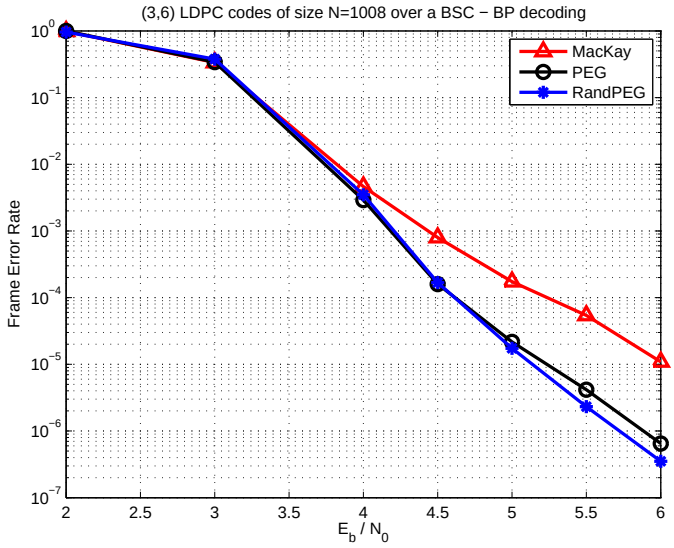


Figure 3.3 – Peg/RandPEG comparison: performance of regular (3,6) LDPC codes of size $N = 1008$ simulated on a BSC channel, and decoded with BP decoding

in linear time with shift registers [PW72]. Finally, hardware decoders can efficiently take advantage of the code structure with parallel decoding [LLT+04]. QC-LDPC have attracted much attention because they are hardware friendly, and most standards that rely on LDPC codes use QC-LDPC codes. Some examples include DVBS2 [DVB05],

WiFi IEEE 802.11n [Wif06], WiMax IEEE 802.16e [WiM05], IEEE 802.3an (10BASE-T) or GSFC-STD-9100 (used by NASA Goddard Space Flight Center) [GSF06]. We first introduce protographs, and then present QC-LDPC codes as a particular form of protograph-based LDPC codes. Then we investigate the potential of the RandPEG algorithm to construct good QC-LDPC codes.

3.3.1 LDPC codes constructed from protographs

Protograph based codes, introduced in [Tho03] are a class of structured LDPC codes (in opposition with random LDPC codes). A *protograph* (projected graph) is a very small bipartite graph with few variable and check nodes, which is associated with a very small adjacency matrix that can have integer coefficients, because parallel edges are allowed in the protograph.

An LDPC code can be constructed by *lifting* the protograph. Lifting a protograph by a factor m means that each element of the small ardency matrix is replaced by a $m \times m$ matrix: a 0 is replaced by an all-zero $m \times m$ matrix, and a '1' is replaced by a permutation matrix. The lifted graph is also called an m-cover in graph theory. Lifting the protograph is done in such a way that multiple edges are not allowed.

The protograph describes the structure of the code, and can be used as a detailed representation of the lifted graph for an asymptotic analysis. Indeed, one important property of protograph based codes not shared by other classes of irregular codes is that the local neighborhood of a node is completely determined by the protograph. Hence, one irregularity profile can have several protograph representations, and several decoding thresholds associated. In fact, for a given irregularity profile, a protograph based code is in a subset of the random ensemble code, since interleaver between variable nodes and check nodes is not random, but constrained by the projected graph.

Moreover, the detailed representation allows to control the connectivity of degree 2 variable nodes, and therefore to design codes with good decoding thresholds *and* a minimum distance that grows linearly with the block length. [DJDT05, DDJ06]. Similarly, the authors give in [ASRD07] some asymptotic results that allow to determine whether or not the typical smallest trapping set size grows linearly with codeword length. The asymptotic results are derived from the finite-length results by letting the block length go to infinity.

The protograph approach has enabled the design of very efficient rate compatible codes well adapted to hardware implementation [Dol05] The design of short length protograph-based codes is addressed in [DDJ07] by designing a protograph with very low variable node degrees.

3.3.1.1 Protograph structures

It is possible to impose a particular code structure to the protograph, and various accumulator based structures where proposed: Repeat Accumulate [DJM98], Irregular Repeat Accumulate (IRA), introduced in [JKM00] and thoroughly studied in [RGCV04]. IRA protograph based codes are called Structured IRA (S-IRA) codes, and can be encoded in linear time with respect to the codeword length. It is possible to introduce a second accumulator (Accumulate Accumulate) which results in S-IRAA codes [LSL$^+$06]. Another structure is to accumulate first, which gives rise to the Accumulate Repeat Accumulate (ARA) [ADY04a, ADY04b], and a structure with 3 accumulators was proposed in [DDT04]. Depending on the place of the accumulator, these structures have the advantage of a lower decoding threshold and/or a lower error floor (due to a higher minimum distance). Moreover, some structures can be encoded in linear time with respect to the codeword length. Often, the performance improvements in the waterfall region (lower threshold) is obtained at the cost of a higher average number of decoding iterations (slow decoding convergence). The reader is referred to [LSL$^+$06] and references therein for a detailed presentation of accumulator based structures and their respective advantages and disadvantages.

3.3.1.2 Optimization of a protograph

The optimization of a protograph can be done by several means. One approach is based on a simulated annealing algorithm [Tho03, TAD04, ADDT04]. First, the code rate determines the number of variable nodes and check nodes, and a population of initial protographs with random edges is generated. For each protograph, edges connecting the vertices of the protograph are randomly added, removed or swapped, and the corresponding threshold is computed with DE (under Gaussian approximation for complexity issues). This algorithm converges to a protograph with an optimized decoding threshold.

The second approach for optimizing a protograph based on the multi-edge approach [RU04], which is discussed in the following section.

3.3.2 Multi-edge LDPC codes

The multi-edge is a detailed representation of an LDPC code. With detailed representation, it is possible to introduce a fraction of variable nodes of degree one, which has a positive effect on the decoding threshold[3]and punctured nodes (hidden variable nodes).

Contrary to irregular LDPC codes, where all edges are statistically equivalent, there are different edge types in a multi-edge approach: a node of degree d is assigned d

[3]degree one variable nodes are not possible with random (non structured) LDPC code constructions since the corresponding codes are not stable

sockets, and a type (color) is assigned to each socket . Let t denote the total number of socket types. Each variable (check) node is associated to a vector D of length t, the i-th element of which is the number of sockets of type i assigned to it, and the type of a variable (check) node is determined by its vector D.

An example of a multi-edge structure taken from [RU04, Table 8] is presented in Fig. 3.4.

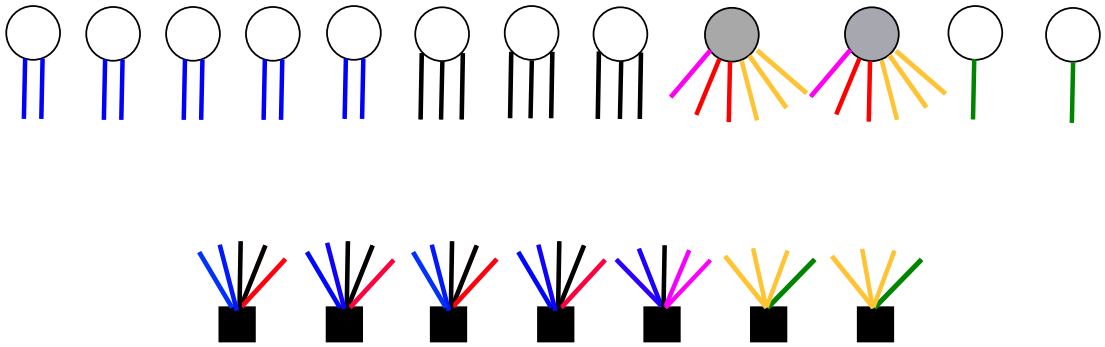


Figure 3.4 – Tanner graph of a multi-edge structure. Unfilled circles represent transmitted variable nodes, filled circles represent hidden ones, and squares represent parity-check nodes. There are 6 socket types (colors)

Let the vector D be blue, black, orange, red, purple, green. Then variable nodes in Fig. 3.4 have four types, namely:
{2, 0, 0, 0, 0, 0}, {0, 3, 0, 0, 0, 0} , {0, 0, 3, 2, 1, 0} and {0, 0, 0, 0, 0, 1}. Check nodes have three types: {0, 0, 3, 0, 0, 1} , {2, 1, 0, 0, 1, 0} and. {2, 2, 0, 1, 0, 0}.

Note that the standard irregular LDPC code ensemble is a particular case of the multiedge structure when all variable nodes are transmitted over the channel and there is only one edge type, *i.e.* any variable node can be connected with any check node when the random permutation of edges is performed.

The protograph ensemble corresponds to a particular case of multi-edge LDPC ensembles when the node and edge types are determined by nodes and edges in its protograph. For a given code structure (the type of nodes and number of sockets is given), it is possible to optimize the proportion of each node type with a multi-edge approach.

3.4 Construction of QC-LDPC codes

A protograph, whether optimized with SA or with a multi-edge approach, must be lifted to design an LDPC code. When lifting a protograph to obtain an LDPC code, one can restrict the set of permutation matrices to the set of circulant permutation matrices,

also called circulants[4]. The resulting code is then quasi-cyclic [Oka03, Fos04], and have a very low desciptional complexity compared to randomly lifted codes.

EG/PG codes can be put in quasi-cyclic form. The parity-check matrix of such codes is square (but not full rank !) and there are many redundant lines. As a consequence, pseudo distance equals the minimum distance, which is a remarkable property.

In [Fos04], necessary conditions to have a QC-LDPC code of girth 6,8,10,12 are given, and two different approaches are investigated for the construction of QC-LDPC codes: a random search approach, and a structured search approach.

3.4.1 Structured search of QC-LDPC codes

In this section, we consider a class of regular QC LDPC codes. We are interested in constructing QC-LDPC graphs of a given target girth g_t. Let (d_v, d_c) denote a regular LDPC of row weight d_c and column weight d_v, and N denote size of the code (codeword length). Moreover, let m denote the size of the circulant permutation matrix used for the lifting.

The parity-check matrix H can be written:

$$H = \begin{bmatrix} I(p_{1,1}) & I(p_{1,1}) & \dots & I(p_{1,d_c}) \\ I(p_{2,1}) & I(p_{2,1}) & \dots & I(p_{2,d_c}) \\ \vdots & & \ddots & \vdots \\ I(p_{d_v,1}) & I(p_{d_v,1}) & \dots & I(p_{d_v,d_c}) \end{bmatrix}$$

where I_p denotes a circulant permutation matrix obtained by shifting the identity p times to the right. Thus, I_0 denotes the identity matrix.

In [Fos04], the shifting coefficients $p_{i,j}$ are determined with two different approaches. The first is random assignment: all the coefficients are assigned at random, until a graph of girth g_t is found. This method not computationally efficient. The second approach is a structured search, and consists of defining an application $p : (i, j) \rightarrow p_{i,j}$, and three explicit structured search applications were proposed.

- $p_{j,l} = jq_1 + lq_2 \mod m$
- $p_{j,l} = jl \mod m$
- $p_{j,l} = q_1^j + q_2^l \mod m$

3.4.2 Proposed structured search

We propose another structured search function that allows to construct QC-LDPC codes of girth 8. For all the values of d_c that we tested, a column-weight two QC-

[4] A circulant matrix is a shifted identity matrix

LDPC graph (*i.e.* a $(2, d_c)$ graph, also called cycle Tanner graph) of girth 8 can be constructed with the following parameters

- $m \geq d_c + 1$
- $p_{j,l} = (l - j)^2$

We point out that the value $m = d_c + 1$ matches the lower bound given in [Fos04]. Specifically, it can be shown that $(2, dc)$ graphs lifted with a circulant size $m < d_c + 1$ cannot have girth 12. This means that, similarly to the RandPEG algorithm, this construction with $m = d_c + 1$ enables to design quasi-cyclic cages.

We used the same structured search to design QC-LDPC codes of girth 12. Only some values of the circulant size give rise to a graph of girth 12. Such constructions are given in appendix C.

3.4.3 QC-RandPEG

In [LK04], the authors propose to construct QC-LDPC codes based on a modified PEG algorithm that adds a quasi-cyclic constraint in the construction algorithm. In [ADDT04], the authors mention that the lifting of of QC graphs is achieved with a modified version of the PEG algorithm that they call *circulant PEG* (cPEG), that enforces the circulant condition. Similarly, it is possible to use the RandPEG algorithm to construct QC-LDPC codes. We first discuss the importance of multiple-stage lifting.

3.4.3.1 Multiple-stage lifting

For large graphs, it is important to lift a protograph to the full graph in multiple stages, that is to say by using the expansion algorithm recursively. Fig. 3.5 shows the parity-check matrix of a SIRA code lifted in one stage with a lifting factor $m = 1024$. Fig. 3.6 shows the parity-check matrix of the same SIRA protograph lifted in two stages, with respective lifting factors $m_1 = 4$ and $m_2 = 256$.

A first lifting will separate parallel edges. Let us now discuss two main arguments that show the importance of multiple-stage lifting.

Number of configurations With multiple-stage lifting, the number of possible configurations is much larger, as pointed out with the following example. The fact that the number of configurations is large is important for two reasons. First the probability that a good configuration exists in the set is more important. Second, the probability that the QC-RandPEG finds a good configuration is more important.

Let N be the number of non zero elements in the protograph matrix counted with their multiplicity (*i.e.* the number of circulants to choose for the lifting). Let $m =$

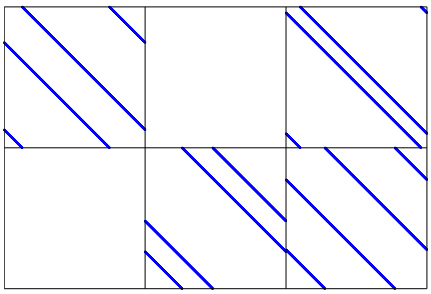


Figure 3.5 – SIRA protograph with 1-stage lifting. Lift size $m = 1024$

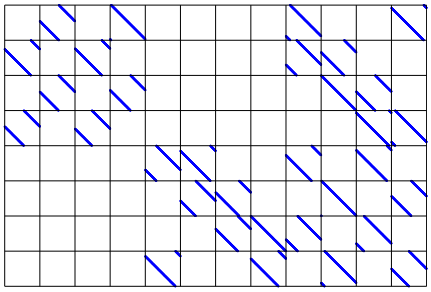


Figure 3.6 – SIRA protograph with 2-stage lifting. Total lift size $m = 1024 = 4 \times 256$

$m_1 m_2$ be the lifting parameter. Lifting the SIRA protograph with a single m-lift gives m^N different configurations, because for each lifted bloc, we choose 1 circulant among m. With double lifting, say a first m_1-lift followed by an m_2-lift, we have m_1^N configurations for the first lifting, which gives $m_2^{(m_1^N)}$ different configurations after the second lift. When first lifting is not restricted to circulant permutations, then the number of configurations is even larger because for each lifted bloc we choose 1 permutation among $m_1!$.

Inevitable cycles It was shown in [Fos04] that when the base matrix has no zero elements then the maximum girth for such a lifted graph is 12 (*i.e.* there are 12-cycles in the lifted graph).

In [KNCS07a, KNCS07b] the authors generalized this result by showing that lifting a protograph of girth $2g$ gives rise to a lifted graph of girth $6g$, regardless from the shift

values of the circulants.[5] The cycles of length $6g$ are called *inevitable cycles*.

With multiple lifting, the first lifting can greatly reduce the number of inevitable cycles. By lifting a first time, one can design a larger protograph with no 4-cycles, and therefore the graph for the second lifting is no longer upper bounded by 12 [KNCS07b].

A graph lifted in one stage represented in Fig. 3.5. Graphs with a single lift are all of girth 8, whereas it was possible to design a graph lifted in two stages of girth 14. The corresponding graph is represented in Fig. 3.6. Extensive tests showed that a one stage lifted graphs exhibit a catastrophic behavior, namely the FER in the error floor is above 10^2 (!).

The simulation results for the two stage lifted graph are presented in Fig. 3.7. Simulation results given in [LSL$^+$06] are also reported for reference, and is labeled 'cPEG'. Note that a direct comparison of these simulation results is not completely relevant since in [LSL$^+$06] the simulation conditions are not given (Decoding algorithm, maximum number of iterations). Nevertheless, it appears that the RandPEG is a good algorithm for lifting protographs.

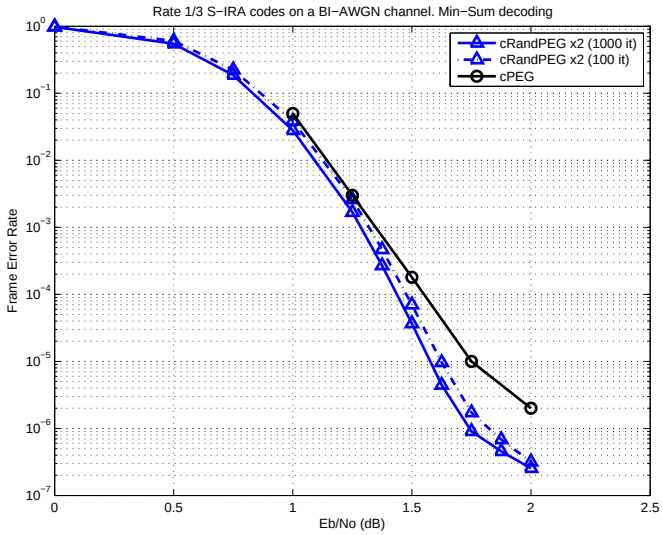


Figure 3.7 – FER versus E_b/N_0 for the SIRA codes in Fig. 3.6 on a BIAWGN channel with a Corrected Min Sum decoding with 100 and 1000 maximum decoding iterations. Results of [LSL$^+$06] are also shown (label 'cPEG').

[5] In [Fos04] the protograph has a girth $g = 4 = 2 \times 2$ and therefore the lifted graph will have inevitable cycles of length $12 = 6 \times 2$.

3.5 Summary

In this chapter, we addressed the construction of LDPC graphs. We have presented an enhanced version of the PEG algorithm, called RandPEG algorithm, based on: 1) the introduction of a larger amount of randomness in the construction, 2) a different manner for spanning and using the construction tree of the original PEG algorithm, and 3) the introduction of an objective function that minimizes the girth multiplicity. The RandPEG greatly improves the achievable girth for given graph parameters, especially for column-weight two graphs ($d_v = 2$) where it allows to construct cages, *i.e.* graphs with minimal size for a given girth. We showed the usefulness of the RandPEG algorithm for constructing NB-LDPC codes.

We extended the RandPEG to construct QC-LDPC graph by lifting protographs. Our simulation results show that the RandPEG algorithm enables to reproduce the results reported in [LSL$^+$06], where the protographs are lifted with an undocumented algorithm.

Finally, we reported a structured search procedure to construct regular QC column-weight two of girth 8. Remarkably, this search construction allows to construct cages.

$g \backslash d_c$	3	4	5	6	7	8	9	10	$\cdots$	50
6	6^* [6]	10^* [10]	15^* [15]	21^* [21]	28^* [28]	36^* [36]	45^* [45]	55^* [55]	$\ldots{}^*$	1275^* [1275]
8	9^* [9]	16^*[20]	25^* [35]	36^* [48]	49^* [70]	64^* [116]	81^* [162]	100^* [230]	$\ldots{}^*$	2500^* [???]
10	15^* [18]	$38^{(34)}$[42]	$90^{(65)}$ [110]	$189^{(111)}$ [225]	$385^{(175)}$ [441]	$728^{(260)}$ [812]				
12	21^* [27]	52^* [104]	$105^{*\dagger}$ [380]	$186^{*\dagger}$ [966]						
14	36^* [36]	260 [292]								
16	45^* [72]	$160^{*\dagger}$ [850]								
18	$114^{(69)}$ [150]									
20	$201^{(93)}$ [285]									
22	$447^{(141)}$ [558]									

Table 3.4 – For various values of girth g and various values checknode degree d_c, we report the smallest graph size N such that the RandPEG algorithm could construct a regular $(2, d_c)$ graph of girth g. When this value achieves the lower bound $N_g^{(2,d_c)}$, we indicate so by super-scripting with a star $(*)$, thus indicating that the graph is minimal. Otherwise the value of the lower bound $N_g^{(2,d_c)}$ is super-scripted with parenthesis. Some values are super-scripted with a dag, which means that the RandPEG was initialized with a tree for these constructions. For comparison, the value of N such that the standard PEG algorithm could construct the corresponding graph is reported in square brackets.

Conclusion and perspectives

Conclusions

We first presented an analytical asymptotic analysis of jointly decoded Raptor codes over a BIAWGN channel, based on Information Content evolution. Based on the analysis, we derived an optimization method for the design of efficient output degree distributions, and thoroughly studied the influence of the optimization parameters. The optimization problem is linear with respect to the coefficients of the output degree distribution and therefore can efficiently be solved with linear programming. We showed that even though Raptor codes are not universal on other channels than the BEC, Raptor codes optimized for a given channel capacity also perform well on a wide range of channel capacities when joint decoding is considered. We also investigated the extension of the analysis to the uncorrelated Rayleigh fading channel with perfect CSIR. In our rateless setting, the optimization of Raptor codes for quasi-static fading channels consists in optimizing a degree distribution that operates close to the channel capacity *simultaneously* for different channel capacities. We showed that it is possible to constrain the optimization problem such that convergence of the decoder is enforced simultaneously for different channel capacities. This allowed us to propose new distributions, which show great robustness to channel variation.

We also investigated the design of efficient finite length Raptor codes. Specifically, we proposed a rate splitting strategy and showed that the use of lower rate LDPC precodes ($R_p \simeq 0.9$) is a valid strategy for designing Raptor codes that perform well at small to moderate lengths on various channels. In particular, we did not consider concatenation with Hamming codes for the precode, because these constructions are specifically designed for the BEC case. Interestingly, our results show that with our

approach, the use of Hamming codes is not necessary even for the BEC.

Finally, we investigated the behavior of various distributions over higher order modulations. Even though the degree distributions were optimized for BPSK modulation, we showed with simulation results that they perform well on higher order modulations, for both the AWGN channel and the uncorrelated Rayleigh fading channel. We showed that in presence of imperfect CSIR, it is possible to improve the performance with no additional complexity, by using an appropriate metric for the computation of the LLR at the output of the channel.

In the second part of this thesis, we proposed the RandPEG algorithm, which is an enhanced version of the PEG algorithm. The proposed algorithm greatly improves the girth properties of the resulting graphs: given a fixed graph size, the achievable girth g is increased, or when the girth cannot be increased, our modified algorithm minimizes the number of cycles of length g. The RandPEG algorithm was successfully used to design efficient LDPC (pre)codes for the design of efficient Raptor codes. We also presented both a structured search procedure and an extension of the RandPEG algorithm to construct QC-LDPC codes with good girth properties,

Perspectives

Several points constitute interesting perspectives to the work presented in this thesis.

- In the rateless framework proposed in [CM06a], each early successful decoding means that some extra channel uses become available for the next transmissions, and the authors show that reliable communication is possible on quasi-static and bloc fading channels. In that context, it is interesting to see to what extent carefully optimized Raptor codes constitute an efficient rateless scheme for block fading channels.
- Extension to the BSC has great potential from a practical point of view. Indeed, rather that discarding packets with a faulty checksum at the physical/link layer, the packets could be presented to the above layer, and errors corrected in the layers above. Therefore, Raptor codes that perform well on a BSC when decoded with a low complexity decoder (such as the Erasure Decoder) could advantageously be used with a cross layer approach. The optimization for this specific channel and decoder combination deserves further investigation.
- We have not optimized output degree distributions for high order modulations. Taking into account the demapper and optimizing specifically for the density of messages at the demapper output can bring some substantial performance improvement.

Concerning the design of finite length LDPC codes

- Even though the RandPEG is very efficient for the design of column-weight two graphs, it remains strongly suboptimal for the design for graphs with column-weight $d_v \geq 3$. An efficient algorithm for the design of $d_v \geq 3$ graphs with optimal girth remains a challenge.
- The structured search proposed for the design of QC-LDPC codes has empirically proved to be efficient for the design of column-weight two graphs of girth 8. However, we have no mathematical evidence on the conditions under which the construction indeed gives a graph of girth 8.

A

Proof of proposition 2.3

Let F be defined by: $F = (\psi \circ \phi)$, where ϕ is defined by equation (2.11) $x_v^{(l)} = \phi(x_u^{(l-1)})$:

$$\phi(x) = \sum_{i=1}^{d_v} \iota_i J\Big((i-1)J^{-1}(x) + \tau(x)\Big) \tag{A.1}$$

with

$$\tau(x) = J^{-1}\Big(T\Big(\sum_{i=1}^{d_v} I_i J\big(iJ^{-1}(x)\big)\Big)\Big) \tag{A.2}$$

and ψ is defined by the Check Node message update equation (2.12) $x_u^{(l)} = \psi(x_v^{(l)})$:

$$\psi(x) = 1 - \sum_{j=1}^{d_c} \omega_j J\Big((j-1)J^{-1}(1-x) + f_0\Big) \tag{A.3}$$

It suffices to prove the following result: $\lim_{x\to 0} F'(x) = \omega_2 \alpha e^{-f_0/4}$

First we give mention that $J(0) = 0$, $J'(0) \neq 0$. Moreover, $T'(0) = 0$ for an LDPC precode where $\rho_2 = 0$ which is always true for practical codes, and simple calculus gives $\tau'(0) = 0$. First we compute $\phi'(0)$:

$$\phi'(x) = \sum_{i=1}^{d_v} \iota_i\big((i-1)(J^{-1})'(x) + \tau'(x)\big)J'[(i-1)J^{-1}(x)]$$

$$\lim_{x\to 0} \phi'(x) = \lim_{x\to 0} \sum_{i=1}^{d_v} \iota_i (i-1) \frac{J'((i-1)J^{-1}(x))}{J'(J^{-1}(x))} = \sum_{i=1}^{d_v} \iota_i (i-1) = \alpha$$

Then, we compute $\psi'(0)$: $\psi'(x) = \sum_{j=1}^{d_c} \omega_j (j-1)(J^{-1})'(1-x)J'[(j-1)J^{-1}(1-x)+f_0]$
Let μ be defined by $\mu = \mu(x) = (J^{-1})(1-x)$. Then, we obtain:

$$\psi'(x) = \sum_{j=1}^{d_c} \omega_j (j-1) \frac{J'[(j-1)\mu + f_0]}{J'(\mu)}$$

Then, using the following approximation of $J'(\mu)$ for μ given in [RGCV04]: $J'(\mu) \sim \log_2(e)\frac{\sqrt{\pi}e^{-\mu/4}}{4\sqrt{\mu}}$

$$\begin{aligned}\lim_{x\to 0} \psi'(x) &= \lim_{\mu\to\infty} \sum_{j=1}^{d_c} \omega_j (j-1) \frac{J'[(j-1)\mu + f_0]}{J'(\mu)} \\ &= \lim_{\mu\to\infty} \sum_{j=1}^{d_c} \omega_j (j-1) \sqrt{\frac{\mu}{(j-1)\mu + f_0}} e^{-\frac{(j-2)\mu+f_0}{4}} = \quad \omega_2 e^{\frac{-f_0}{4}}\end{aligned}$$

Finally, $\phi(0) = 0$, and $F'(x) = \phi'(x)(\psi' \circ \phi)(x)$ gives $\lim_{x\to 0} F'(x) = \alpha\omega_2 e^{\frac{-f_0}{4}}$
□

B

Asymptotic analysis of Raptor codes on the BEC

B.1 Density evolution

Because raptor codes are linear codes and the BEC is a symmetric channel, we can assume, without loss of generality, that the all-zero codeword has been transmitted. In that case, the messages on the edges of the decoding graph are 1 and 0, where the value 0 indicates an erasure: the value is 0 iff the corresponding edge is connected to an input symbol that has not been recovered.

When the precode is an LDPC code with data node and check edge distributions $\Lambda(x)$ and $\rho(x)$, its extrinsic transfer function [Bri01] is given by:

$$T(x) = 1 - \Lambda\big(1 - \rho(x)\big) \tag{B.1}$$

Similarly to section 2, we assume the reinitialization of the decoder, which is a pessimistic assumption because we under-estimate the information provided by the precode. However, this pessimistic assumption is crucial to lead to a *linear* optimization problem, and proves sufficient for the design of efficient output degree distributions.

We denote $p^{(l)}$ (resp. $q^{(l)}$) the probability that an edge connecting a dynamic check node to an input symbol (resp. an input symbol to a dynamic check node) carries the value 1 at the l^{th} decoding iteration. We denote by $u^{(l)}$ the extrinsic information passed by the LT code to the precode, at the l^{th} decoding iteration. As the input symbols are

of average degree α, we have:

$$v^{(l)} = T(u^{(l)}) = T\big(1 - e^{-\alpha q^{(l-1)}}\big) \tag{B.2}$$

The extrinsic information passed by the precode to the LT code is then $v^{(l)} = T(u^{(l)})$. When accounting for the transfer function of the precode, the update rules for the messages in the Tanner graph can be written as follows:

$$p^{(l)} = 1 - (1 - v^{(l)})e^{-\alpha q^{(l-1)}} \tag{B.3}$$

$$1 - q^{(l)} = \omega(p^{(l)}) \tag{B.4}$$

$$q^{(l)} = F(q^{(l-1)}) = \omega\left(1 - e^{-\alpha q^{(l-1)}}\left(1 - T\big(1 - e^{-\alpha q^{(l-1)}}\big)\right)\right) \tag{B.5}$$

Combining (B.3), (B.4) and (B.2) gives (B.5), that describes the evolution through one joint decoding iteration of the probability of erasure at the output of the dynamic check nodes. Note that for a given distribution $\iota(x)$, this expression is linear with respect to the coefficients of $\omega(x)$, which is the distribution that we intend to optimize. We point out that (B.5) is general since it reduces to the classical sequential decoding case by setting the extrinsic transfer function to $x \mapsto T(x) = 0 \quad \forall x \in [0; 1]$, thus assuming that no information is propagated from the precode to the fountain.

B.2 Fixed point characterization

In a density evolution analysis, the convergence is guaranteed by $F(x) > x$, and the convergence continues toward a fixed point of $x \mapsto F(x)$. Unfortunately, there are no trivial solutions for the fixed point of (B.5). However, using a functional analysis, an upper bound on the fixed point can be given. Replacing q by 1 and using the fact that $T(1) = 1$ in (B.5), we obtain:

(B.6)

which means that because $x \mapsto F(x)$ is an increasing function, the fixed point is necessarily less or equal than 1.

This result gives some insights why small decoding probability cannot be achieved with a constant average degree $\Omega'(1)$, which is a well known fact [Lub02]. Loosely speaking, if the distribution is capacity achieving, then we have a posteriori rate $\frac{\Omega'(1)}{\alpha} = 1$. A decoding error probability δ, becomes arbitrarily small when $\alpha \to \infty$, meaning that $\Omega'(1)$ also grows without bounds.

In fact, it can be shown that the average degree of the output symbols has to grow at least logarithmically with K, the number of input symbols. More precisely, a *reliable* decoding algorithm is a decoding algorithm that can decode K input symbols from

any N collected output symbols, with an error probability that is at most $1/K^c$ for some positive constant c. If the decoding has reached its fixed point, the the fraction of unrecovered symbols is $e^{-\alpha}$. The result is then given by the fact that

$$e^{-\alpha} < \frac{1}{K^c} \Longleftrightarrow \alpha > c\log(K)$$

and that

$$\alpha = (1+\epsilon)\Omega'(1)$$

B.3 Starting condition

At the first iteration, $q^{(0)} = 0$. Therefore, according to (2.11), $p^{(1)} = 0$. Reporting this in (B.5) gives:

$$q^{(1)} = F(0) = \omega_1$$

Lemma B.1. *The decoding process can begin iff $q^{(1)} > \varepsilon$, for some arbitrary $\varepsilon > 0$, which gives:*

$$\omega_1 > \varepsilon \tag{B.7}$$

Therefore, one must have $\omega_1 > 0$ for the decoding process to begin, and ε appears to be a design parameter that will constrain the optimization problem, ensuring that $\omega_1 \neq 0$. In practice, the value of ε can be chosen arbitrarily small.

As an illustration we point out that, for the "Ideal Soliton Distribution" introduced by Luby [Lub02], $\Omega_1 = 1/K$, which is the smallest proportion possible with K input symbols.

B.4 Lower bound on the edge proportion of degree 2 output symbols

Following the same approach developed for the BIAWGNC case, we derive the following lower bound on ω_2:

Lemma B.2. *For an output degree distribution that is to be capacity achieving, we have:*

$$\omega_2 > \frac{1}{\alpha} \tag{B.8}$$

Proof: by derivating $x \mapsto F(x)$ defined in (B.5), we get: $\lim_{x\to 0} F'(x, T(.)) = \alpha\omega_2$. Moreover, for a capacity achieving degree distribution, $\omega_1 = 0$ [ES06], which means that $F(0) = 0$. Then, the convergence condition $F(x, T(.)) > x$ implies that $\lim_{x\to 0} F'(x, T(.)) > 1$, which gives the result.

This inequality can also be written in terms of the *node* proportion of output symbols of degree 2: $\Omega_2 > \frac{\Omega'(1)}{2\alpha}$. Moreover, for a capacity achieving distribution the code rate $\frac{\Omega'(1)}{\alpha}$ would be equal to one, which gives the well known result: $\Omega_2 \geq \frac{1}{2}$ for a capacity achieving distribution.

B.5 Optimization problem statement

The optimization of an output distribution consists in maximizing the a posteriori rate of the corresponding LT code, *i.e.* maximizing $\Omega'(1) = \sum_i \Omega_i i$, which is equivalent to minimizing $\sum_i \omega_i / i$. Moreover, according to the previous section, several constraints must be satisfied.

[C_1] Since $\omega(x)$ is a probability distribution, its coefficients must sum up to 1. We call this the *proportion constraint* C_1.

[C_2] To ensure the convergence of the iterative process, we must have $F(x, T(.)) > x$. However, this inequality cannot hold for each and every value of x: the fixed point of $F(x, T(\cdot))$ is smaller than 1. Therefore, we must fix a margin $\delta > 0$ away from 1, and then by discretizing $[0; 1 - \delta]$ and requiring inequality to hold on the discretization points, we obtain a set of inequalities that need to be satisfied: they define the *convergence constraint* C_2.

[C_3] The starting condition (lemma 2.1) must also be satisfied and defines the *starting constraint* C_3.

[C_4] The edge proportion of output symbols of degree 2 is lower bounded by lemma B.8, defining the *stability constraint* C_4.

Finally, $x \mapsto T(x)$ is defined according to (B.1) for an LDPC code, or could be estimated with Monte Carlo simulations if another component code is used as a precode. The IC transfer function $T(\cdot)$ appears in constraint [C_2].

For a given value of α, the cost function and the constraints are linear with respect to the unknown coefficients ω_i. Therefore, the optimization of an output degree distribution can be written as a linear optimization problem that can be efficiently solved with linear programming. The optimization problem can be stated as follows:

$$\omega_{opt}(x) \quad = \quad \arg\min_{\omega(x)} \sum_j \frac{\omega_j}{j} \tag{B.9}$$

subject to the constraints:

[C_1] Proportion constraint: $\sum_i \omega_i = 1$

[C_2] Convergence constraint: $F(x, T(.)) > x \quad \forall x \in [0; 1-\delta] \quad$ for some $\delta > 0$

[C_3] Starting condition: $\omega_1 > \varepsilon \quad$ for some $\varepsilon > 0$

[C_4] Flatness condition: $\omega_2 > \frac{1}{\alpha}$

B.6 Finite length design

B.6.1 "Asymptotic design" for finite length distributions

One might question whether the asymptotic analysis of the joint decoder is of any interest for finite length design. In the asymtotic regime, *i.e.* when the codeword length is infinite, the decoding trajectory in the EXIT chart will fit between the curves $y = x$ and $y = F(xT(\cdot))$.

The $x \to F(x, \sigma^2, T(\cdot))$ characterizes the expectation of the decoder in the asymptotic regime. In the asymptotic regime, the concentration theorem [LMSS01b] ensures that the performance of a randomly sampled code converges to the expected performance as the codeword length increases.

However, the concentration to the expected performance does not hold for the finite length case, and one must account for a some variance in the decoding trajectories. Following the steps of [Sho06], we use the following convergence constraint in the optimization problem for finite length:

[C_2] Convergence constraint:

$$F(x, \sigma^2, T(.)) > x + \frac{c}{K}\sqrt{1-x} \quad \forall x \in [0; x_0 - \delta] \quad \text{for some } \delta > 0$$

B.6.2 Simulation results

We optimized output degree distributions for 4 different precodes with different rates. We restricted ourselves to regular LDPC precodes because for high rates, regular codes are known to have good thresholds, close to the irregular thresholds.

Figures B.6.2, B.6.2, B.6.2 and B.6.2 show simulation results for Raptor codes of length $K = 1024, 2048, 4096$ and 8192 respectively, constructed with precodes described in section 2.4.2.

In fact, according to the cycle spectrum given in section 2.4.2, it appears that all curves that exhibit an error floor are associated with a precode with cycles of length 4. This shows that as long as joint optimization using the precode transfer function is performed, a lower rate precode does not impact the performance of the Raptor code in the waterfall region. Note hat none of our simulations with the lowest precode rate $R_p = 0.9$ show error floors, despite the fact that we do not use Hamming codes.

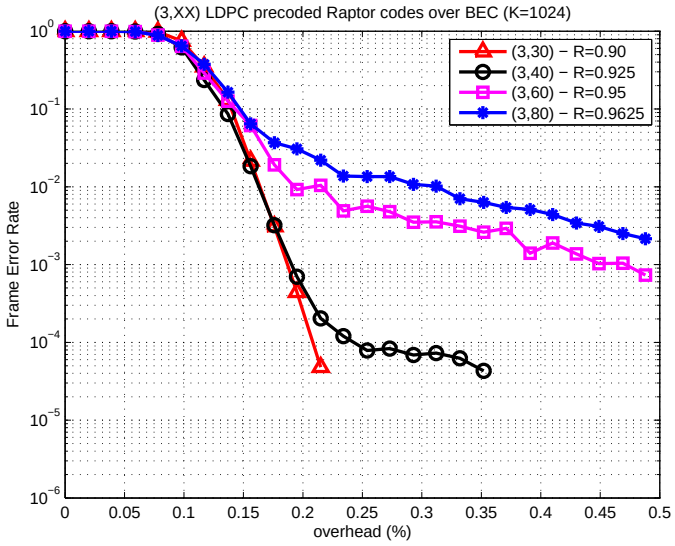


Figure B.1 – Performance over a BEC of LDPC precoded Raptor codes of size $K = 1024$. Only the lowest considered rate $R = 0.9$ shows good performance. For all other precode rates, the code exhibits an error floor behavior.

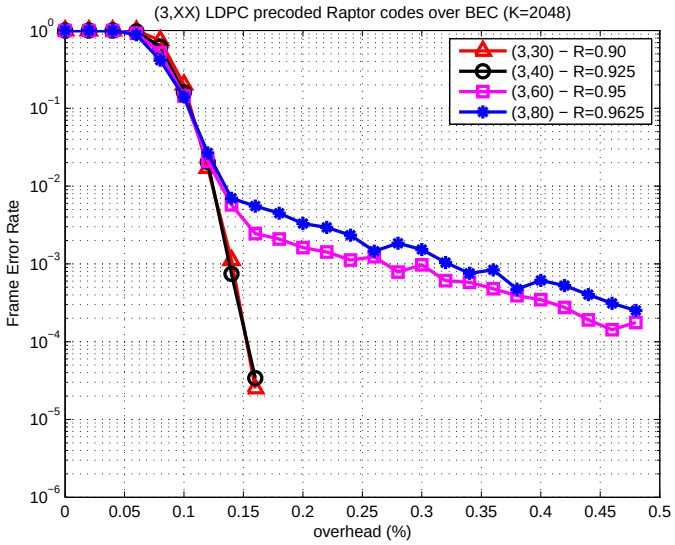


Figure B.2 – Performance over a BEC of LDPC precoded Raptor codes of size $K = 2048$. The lowest rates ($R = 0.9$ and $R = 0.925$) show good performance, whereas the Raptor codes with precodes of higher rates exhibit an error floor.

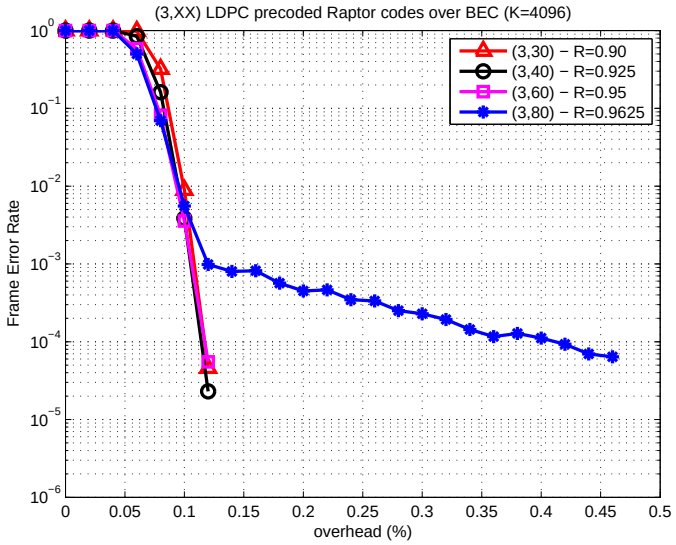


Figure B.3 – Performance over a BEC of LDPC precoded Raptor codes of size $K = 4096$. Only the precode of highest rate $R = 0.9625$ exhibit an error floor behavior

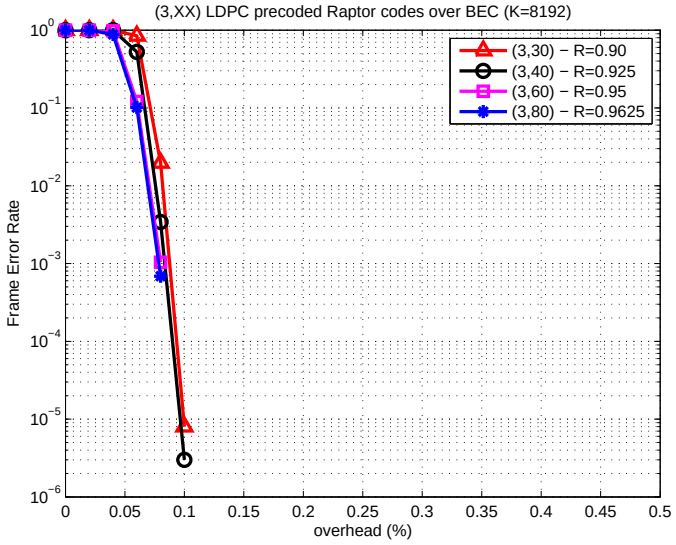


Figure B.4 – Performance over a BEC of LDPC precoded Raptor codes of size $K = 8192$. For $K = 8192$, all precodes are of girth 6, the corresponding Raptor codes have similar performance

C

Structured constructions of $(2, d_c)$ QC-LDPC codes of girth 12

We use the same notations as in section 3.4.1. Let (d_v, d_c) denote a regular LDPC of row weight d_c and column weight d_v, and N denote the total length of the code. Moreover, m denotes the size of the circulant permutation matrix used to construct a QC-LDPC code.

Let $\mathcal{G}_g^{(d_v,d_c)}$ denote the lower bound on N such that a (d_v, d_c) code of girth g exists. The bound is computed according to [HEA05]. The following design parameters where found to construct column-weight two QC-LDPC codes of girth 12. The corresponding construction parameters are graphically represented in Fig. C.1.

$(2, 3)$ codes

The bounds are $\mathcal{G}_{12}^{(2,3)} = 21$ and $\mathcal{G}_{14}^{(2,4)} = 33$.

$N = 21$ $(m = 7)$ and $p_{j,l} = (i + j)^{27,28}$
$N = 24$ $(m = 8)$ and $p_{j,l} = 0$ if $j = 0$, l otherwise $(\forall m = 8 - 20)$
$N = 27$ $(m = 9)$ and $p_{j,l} = (l + j)^{\{23,25,26\}}$
$N = 30$ $(m = 10)$ and $p_{j,l} = (l + j)^{\{23,26,27,29,31\}}$
$N = 33$ $(m = 11)$ and $p_{j,l} = (l + j)^{\{7,9,17,19,25-28,32-34\}}$ or $(i - j)^{4,6,8,10,16,18,20,22,28,30}$
$N = 36$ $(m = 12)$ and $p_{j,l} = (l + j)^{\{24,26,28-31\}}$
$N = 39$ $(m = 13)$ and $p_{j,l} = (l + j)^{\{3-5,7-9,11,15-17,19-21,23,26,29-31\}}$
or $(i - j)^{4,6,8,10,16,18,20,22,28,30}$

$(2, 4)$ codes

The bounds are $\mathcal{G}_{12}^{(2,4)} = 52$ and $\mathcal{G}_{14}^{(2,4)} = 106$.

$N = 52$ $(m = 13)$ and $p_{j,l} = (l + j)^{7,19}$
$N = 60$ $(m = 15)$ and $p_{j,l} = 0$ if $j = 0$, l^2 otherwise
$N = 68$ $(m = 17)$ and $p_{j,l} = (l + j)^{4,15,20,21}$ or $(l - j)^{6,12,22,28}$
$N = 76$ $(m = 19)$ and $p_{j,l} = (l - j)^{14}$ or $(i + j)^{17,21}$
$N = 80$ $(m = 20)$ and $p_{j,l} = (i + j)^{26}$
$N = 84$ $(m = 21)$ and $p_{j,l} = 0$ if $j = 0$, $l^{\{3,4\}}$ otherwise, or $(i - j)^{4,10,16,22,28}$
$N = 88$ $(m = 22)$ and $p_{j,l} = 0$ if $j = 0$, $l^{\{2,3\}}$ otherwise
$N = 92$ $(m = 23)$ and $p_{j,l} = (l + j)^{\{3,5,6,10,20,21,26,30\}}$ or $(i - j)^{4,6,10,14,16,20,26,28}$
$N = 100$ $(m = 25)$ and $p_{j,l} = (l + j)^{\{3,7,11,26\}}$
$N = 104$ $(m = 26)$ and $p_{j,l} = 0$ if $j = 0$, $l^{\{2,3\}}$ otherwise or $(i + j)^{7,19,27,31}$
$N = 108$ $(m = 27)$ and $p_{j,l} = 0$ if $j = 0$, l^2 otherwise or $(i - j)^{4,14,16,22}$ or $(i + j)^{24}$
$N = 116$ $(m = 29)$ and $p_{j,l} = (l + j)^{\{3,5,6,9,12,13,18,19\}}$
$N = 172$ $(m = 43)$ and $p_{j,l} = (l + j)^{\{3,4,7,8,9,12,16,17,18,19\}}$
$N = 212$ $(m = 53)$ and $p_{j,l} = (l + j)^{\{3,4,5,7,8,9,10,12,14,16,20\}}$

$(2, 5)$ codes

Bounds : $\mathcal{G}_{12}^{(2,5)} = 105 = 21 \times 5$ and $\mathcal{G}_{14}^{(2,5)} = 255$.

Tested parameters : $(i[+-]j)^q$ for $q = 2 : 31$ and $m = 21 : 51$

$N = 115$ $(m = 23)$ and $p_{j,l} = (l - j)^{28}$ or $(l + j)^{10}$
$N = 125$ $(m = 25)$ and $p_{j,l} = (l + j)^{11}$
$N = 145$ $(m = 29)$ and $p_{j,l} = (l + j)^{9}$
$N = 155$ $(m = 31)$ and $p_{j,l} = (l + j)^{3,5}$
$N = 165$ $(m = 33)$ and $p_{j,l} = (l - j)^{28}$
$N = 175$ $(m = 35)$ and $p_{j,l} = (l + j)^{3,15}$
$N = 185$ $(m = 37)$ and $p_{j,l} = (l - j)^{4,14,22}$ or $(l + j)^{3,6}$
$N = 195$ $(m = 39)$ and $p_{j,l} = (l + j)^{19,21}$
$N = 205$ $(m = 41)$ and $p_{j,l} = (l - j)^{26}$ or $(l + j)^{3,8}$
$N = 215$ $(m = 43)$ and $p_{j,l} = (l - j)^{18,26}$ or $(l + j)^{3}$
$N = 220$ $(m = 44)$ and $p_{j,l} = (l + j)^{27}$
$N = 225$ $(m = 45)$ and $p_{j,l} = (l + j)^{22}$
$N = 230$ $(m = 46)$ and $p_{j,l} = (l - j)^{28}$ or $(l + j)^{10,20,24}$
$N = 235$ $(m = 47)$ and $p_{j,l} = (l - j)^{6,12,16,22,28}$ or $(l + j)^{3,16,17,20}$

$N = 245$ $(m = 49)$ and $p_{j,l} = (l - j)^{4,8,16,20,26,28}$ or $(l + j)^{3,9,16}$
$N = 250$ $(m = 50)$ and $p_{j,l} = (l + j)^{11,23}$

$(2, 6)$ codes

Bounds : $\mathcal{G}_{12}^{(2,6)} = 186 = 31 \times 6$ and $\mathcal{G}_{14}^{(2,6)} = 561$.

Tested parameters : $(i + / - j)^q$ for $q = 2 : 31$ and $m = 21 : 74$

$N = 222$ $(m = 37))$ and $p_{j,l} = (l + j)^{16}$
$N = 246$ $(m = 41)$ and $p_{j,l} = (l + j)^{8}$
$N = 282$ $(m = 47)$ and $p_{j,l} = (l - j)^{16}$
$N = 342$ $(m = 57)$ and $p_{j,l} = (l - j)^{14}$
$N = 354$ $(m = 59)$ and $p_{j,l} = (l - j)^{10}$
$N = 366$ $(m = 61)$ and $p_{j,l} = (l + j)^{7}$
$N = 378$ $(m = 63)$ and $p_{j,l} = (l + j)^{22}$
$N = 390$ $(m = 65)$ and $p_{j,l} = (l + j)^{19}$
$N = 402$ $(m = 67)$ and $p_{j,l} = (l + j)^{8,9,13}$ or $(i - j)^{4,24}$
$N = 408$ $(m = 68)$ and $p_{j,l} = (l - j)^{28}$
$N = 414$ $(m = 69)$ and $p_{j,l} = (l + j)^{8}$ or $(i - j)^{18}$
$N = 420$ $(m = 70)$ and $p_{j,l} = (l - j)^{28}$
$N = 426$ $(m = 71)$ and $p_{j,l} = (l + j)^{4,18}$ or $(i - j)^{6}$
$N = 426$ $(m = 73)$ and $p_{j,l} = (l + j)^{7,15}$ or $(i - j)^{22}$
$N = 432$ $(m = 74)$ and $p_{j,l} = (l + j)^{16}$

$(2, 7)$ codes

Bounds : $\mathcal{G}_{12}^{(2,7)} = 301 = 43 \times 7$ and $\mathcal{G}_{14}^{(2,7)} = 1057$.

Tested parameters : $(i + / - j)^q$ for $q = 2 : 31$ and $m = 43 : 75$

$N = 483$ $(m = 69))$ and $p_{j,l} = (l - j)^{18}$

$(2, 8)$ codes

Bounds : $\mathcal{G}_{12}^{(2,8)} = 456$ and $\mathcal{G}_{14}^{(2,8)} = 1828$.

$N = 744$ $(m = 93)$ and $p_{j,l} = 0$ if $j = 0$, l^7 otherwise
$N = 944$ $(m = 118)$ and $p_{j,l} = 0$ if $j = 0$, l^7 otherwise

$N = 984$ $(m = 123)$ and $p_{j,l} = 0$ if $j = 0$, l^3 otherwise
$N = 1016$ $(m = 127)$ and $p_{j,l} = (l - j)^{22} + 7jl$
$N = 1048$ $(m = 131)$ and $p_{j,l} = (l + j)^{16}$
$N = 1096$ $(m = 137)$ and $p_{j,l} = (l - j)^{14}$
$N = 1112$ $(m = 139)$ and $p_{j,l} = (l + j)^9$

$(2, 9)$ codes

Bounds : $\mathcal{G}_{12}^{(2,9)} = 657 = 73 \times 9$ and $\mathcal{G}_{14}^{(2,7)} = 2961$.

Tested parameters : $(i[+-]j)^q$ for $q = 2 : 31$ and $m = 73 : 185$

$N = 1629$ $(m = 181)$ and $p_{j,l} = (l + j)^{13}$

Girth $= 8/10$
$N = 1395$ $(m = 155)$ and $p_{j,l} = (l + j)^{14}$

$(2, 10)$ codes

Bounds : $\mathcal{G}_{12}^{(2,10)} = 910$ and $\mathcal{G}_{14}^{(2,10)} = 4555$.

$g = 8/10$ $\forall p$ and $p_{j,l} = 0$ if $j = 0$, l^2 otherwise
$N = 2830$ $(m = 283)$ and $p_{j,l} = (l + j)^{11}$ or $(l + j)^{15}$

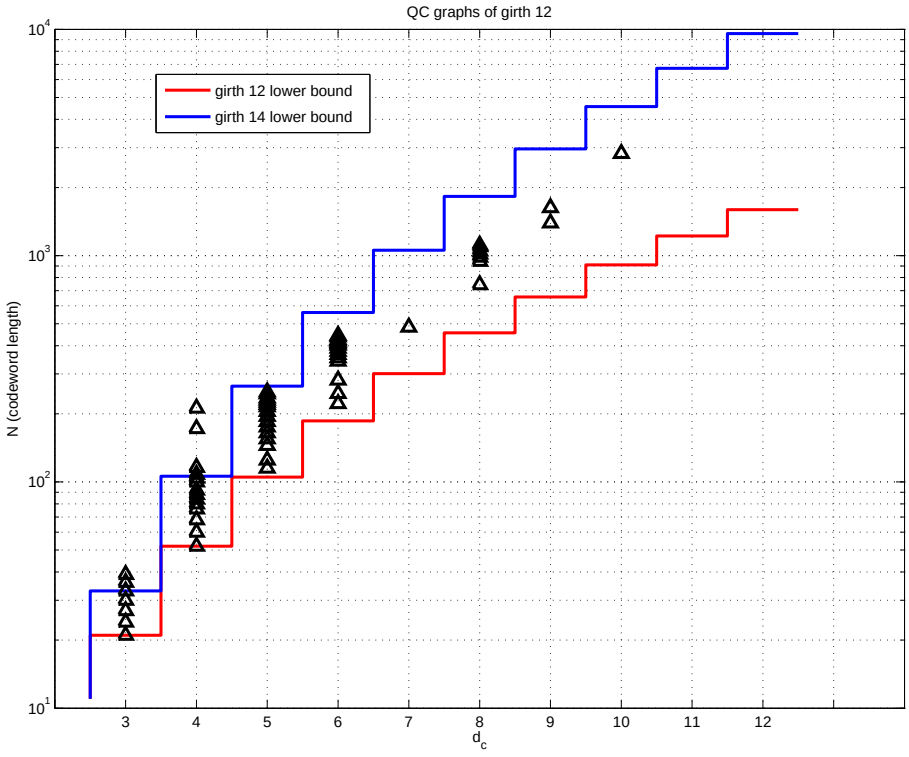


Figure C.1 – For each value of d_c we plot the graph size for which we found constructions cycle graphs $(2, d_c)$ of girth 12 with the structured search procedure. The lower bound on the graph size $\mathcal{G}_{12}^{(2,d_c)}$ (resp. $\mathcal{G}_{14}^{(2,d_c)}$) such that graphs of girth 12 (resp. 14) exist is plotted in red (resp. blue)

Bibliography

[ADDT04] Kenneth Andrews, Sam Dolinar, Dariush Divsalar, and Jeremy Thorpe. Design of low-density parity-check (LDPC) codes for deep-space applications. Technical report, JPL Interplanetary Network Progress (INP) Report 42-159, November 2004.

[ADU02] Abdelaziz Amraoui, Sanket Dusad, and Rudiger Urbanke. Achieving general points in the 2-user gaussian mac without time sharing or rate splitting by means of iterative decoding. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Lausanne, Switzerland*, page 334, June 2002.

[ADY04a] Aliazam Abbasfar, Dariush Divsalar, and Kung Yao. Accumulate repeat accumulate codes. In *Proc. of GLOBECOM, Dallas, Texas*, pages 509–513, November 2004.

[ADY04b] Aliazam Abbasfar, Dariush Divsalar, and Kung Yao. Accumulate repeat accumulate codes. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Chicago, USA*, page 505, July 2004.

[AK04] Masoud Ardakani and Frank R. Kschischang. A more accurate one-dimensional analysis and design of irregular ldpc codes. *IEEE Trans. Commun.*, 52(12):2106–2114, December 2004.

[AK05] Masoud Ardakani and Frank R. Kschischang. Properties of optimum binary message-passing decoders. *IEEE Trans. Inform. Theory*, 51(10):3658–3665, October 2005.

[ASRD07] Shadi Abu-Surra, William E. Ryan, and Dariush Divsalar. Ensemble trapping set enumerators for protograph-based LDPC codes. In *Proc. of 44th Allerton Conf, Illinois, USA*, September 2007.

[BCJR74] L.R. Bahl, J Cocke, F Jelinek, and R Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, 20:284–287, March 1974.

[BGT93] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. Near Shannon limit error-correcting codes and decoding. In *Proc. of International Conference on Communications (ICC), Geneva, Switzerland*, pages 1064–1070, May 1993.

[Big88] Norman Biggs. Constructions for cubic graphs with large girths. *The electronic journal of Combinatorics*, 5(1), 1988.

[BLM02] John Byers, Michael Luby, and Michael Mitzenmacher. A digital fountain approach to asynchronous reliable multicast. *IEEE J. Select. Areas Commun.*, 20(8):1528–1540, October 2002.

[BLMR98] John Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. A digital fountain approach to reliable distribution of bulk data. In *Proc. of ACM SIGCOMM*, pages 56–67, September 1998.

[Bri01] Stephan Ten Brinck. Convergence behavior of iteratively decoded parallel concatenated codes. *IEEE Trans. Commun.*, 49(10):1727–1737, October 2001.

[BRU04] Louay Bazzi, Tom Richardson, and Rudiger Urbanke. Exact thresholds and optimal codes for the binary-symmetric channel and gallager's decoding algorithm A. *IEEE Trans. Inform. Theory*, 50(9):2010–2021, September 2004.

[CFRU01] Sae-Youg Chung, David Forney, Thomas J. Richardson, and Rudiger Urbanke. On the design of low-density parity-check codes within 0.0045dB of the shannon limit. *IEEE Commun. Lett.*, 5(2):58–60, February 2001.

[CM06a] Jeff Castura and Yongyi Mao. On rateless coding over fading channels with delay constraints. In *Proc. of the IEEE Int. Symp. on Inform. Theory (ISIT), Seattle, USA*, July 2006.

[CM06b] Jeff Castura and Yongyi Mao. Rateless coding over fading channels. *IEEE Commun. Lett.*, 10(1), January 2006.

[CRU01] Sae-Youg Chung, Thomas J. Richardson, and Rudiger Urbanke. Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation. *IEEE Trans. Inform. Theory*, 47(2):657–670, February 2001.

[DDJ06] Dariush Divsalar, Sam Dolinar, and Christopher Jones. Construction of protograph LDPC codes with linear minimum distance. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Seatle, Washington, USA*, pages 664–668, July 2006.

[DDJ07] Dariush Divsalar, Sam Dolinar, and Christopher Jones. Short protograph-based LDPC codes. In *Proc. of IEEE MILCOM, Orlando, USA*, pages 1–6, October 2007.

[DDT04] Dariush Divsalar, Sam Dolinar, and Jeremy Thorpe. Accumulate-repeat-accumulate-accumulate codes. In *Proc. of 60th Vehicular Technology Conf. (VTC)*, pages 2292–2296, September 2004.

[Dec] David Declercq. David declercq's homepage. `http://perso-etis.ensea.fr/~declercq/graphs.php`.

[Dec03] David Declercq. *Optimisation et performances des codes LDPC pour des canaux non-standards*. PhD thesis, Univ. Cergy-Pontoise, 2003.

[DF05] David Declercq and Marc Fossorier. Decoding algorithms for non binary LDPC codes over GF(q). *IEEE Trans. Commun.*, 55(4):633–643, April 2005.

[DJDT05] Dariush Divsalar, Christopher Jones, Sam Dolinar, and Jeremy Thorpe. Protograph based LDPC codes with minimum distance linearly growing with block size. In *Proc. of IEEE GLOBECOM, St Louis, MO, USA*, pages 1152–1156, November 2005.

[DJM98] Dariush Divsalar, Hui Jin, and Robert J. McEliece. Coding theorems for 'turbo-like' codes. In *Proc. of 36th Allerton Conf. on Communication, Control, and Computing*, pages 201–210, September 1998.

[Dol05] Sam Dolinar. A rate-compatible family of protograph-based LDPC codes built by expurgation and lengthening. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Adelaide, Australia*, pages 1627–1631, September 2005.

[DPT+02] Changyan Di, David Proietti, I. Emre Telatar, Thomas J. Richardson, and Urbanke Rudiger. Finite length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Trans. Inform. Theory*, 48(6):1570–1579, June 2002.

[DVB05] ETSI Draft EN 302 307 DVBS2. Digital video broadcasting (dvb), March 2005.

[DVB06] DVB-IPDC TS 102 472 v1.2.1 DVBH. Technical specification group services and system aspects, December 2006.

[ES06] Omid Etesami and Amin Shokrollahi. Raptor codes on binary memoryless symmetric channels. *IEEE Trans. Inform. Theory*, 52:2033–2051, May 2006.

[FKLW97] Brendan J. Frey, Frank R. Kschischang, Hans-Andrea Loeliger, and Niclas Wiberg. Factor graphs and algorithms. In *Proc. of 35th Allerton Conf. on Communications, Control and Computing, Monticello, IL*, pages 666–680, September 1997.

[Fos04] Marc Fossorier. Quasi-cyclic low-density parity-check codes from circulant permutation matrices. *IEEE Trans. Inform. Theory*, 50(8):1788–1793, August 2004.

[Gal62] Robert Gallager. Low-density parity-check codes. *IEEE Trans. Inform. Theory*, 8(1):21–28, 1962.

[Gal63] Robert Gallager. *Low-Density Parity-Check Codes*. PhD thesis, MIT, 1963.

[GSF06] STD – 9100 GSFC. Goddard technical standard, May 2006.

[HE03] Xiao-Yu Hu and Evangelos Eleftheriou. Cycle tanner-graph codes over GF(2^b). In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT) 2003, Yokohama, Japan*, page 87, July 2003.

[HEA05] Xiao-Yu Hu, Evangelos Eleftheriou, and Dieter M Arnold. Regular and irregular progressive edge-growth tanner graphs. *IEEE Trans. Inform. Theory*, 51(1):386–398, January 2005.

[HKKM06] Jeongseok Ha, Jaehong Kim, Demijan Klinc, and Steven W. McLaughlin. Rate-compatible punctured low-density parity-check codes with short block lengths. *IEEE Trans. Inform. Theory*, 52(2):728–738, February 2006.

[HM03] Jeongseok Ha and Steven W. McLaughlin. Low-density parity-check codes over gaussian channels with erasures. *IEEE Trans. Inform. Theory*, 49(7):1801–1809, July 2003.

[HM04] Jeongseok Ha and Steven W. McLaughlin. Rate-compatible puncturing of low-density parity-check codes. *IEEE Trans. Inform. Theory*, 50:2824–2836, November 2004.

[HSM01] Jilei Hou, Paul H. Siegel, and Laurence B. Milstein. Performance analysis and code optimization of low-density parity-check codes on rayleigh fading channels. *IEEE J. Select. Areas Commun.*, 19, May 2001.

[HST08] Andrew Hagedorn, David Starobinski, and Ari Trachtenberg. Rateless deluge: Over-the-air programming of wireless sensor networks using random linear codes. In *Proc. of Int. Conf. on Information Processing in Sensor Networks (IPSN), St Louis, MO*, pages 457–466, April 2008.

[JKM00] Hui Jin, Aamod Khandekar, and Robert J. McEliece. Irregular repeat-accumulate codes. In *Proc. of Int. Symp. on Turbo codes and Related Topics*, pages 1–8, September 2000.

[JR06] Hui Jin and Tom Richardson. A new fast density evolution. In *Proc. of IEEE ITW, Punta del Este, Uruguay*, March 2006.

[KFL01] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2):498–519, February 2001.

[KHRM06] Jaehong Kim, Woonhaing Hur, Aditya Ramamoorthy, and Steven W. McLaughlin. Design of rate-compatible irregular ldpc codes for incremental redundancy hybrid ARQ systems. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Seattle, USA*, pages 1139–1143, July 2006.

[KLF01] Yu Kou, Shu Lin, and Marc Fossorier. Low-density parity-check codes based on finite geometries: A rediscovery and new results. *IEEE Trans. Inform. Theory*, 47(4):2711–2736, November 2001.

[KLS04] Richard Karp, Michael Luby, and Amin Shokrollahi. Finite length analysis of LT codes. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Chicago, USA*, July 2004.

[KNCS07a] Sunghwan Kim, Jong-Seon No, Habong Chung, and Dong-Joon Shin. Cycle analysis and construction of protographs for qc ldpc codes with girth larger than 12. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Nice, France*, pages 2256–2260, June 2007.

[KNCS07b] Sunghwan Kim, Jong-Seon No, Habong Chung, and Dong-Joon Shin. Quasi-cyclic low-density parity-check codes with girth larger than 12. *IEEE Trans. Inform. Theory*, 53(8):2885–2891, August 2007.

[LF05] Luis López and Antonio Fernández. A game theoretic analysis of protocols based on fountain codes. In *Proc. of 10th IEEE Symposium on Computers and Communications (ISCC)*, pages 625–630, June 2005.

[LK04] Zongwang Li and B.V.K.Vijaya. Kumar. A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph. In *Proc. of 38th Asilomar Conference on Signals, Systems and Computers (ACSSC)*, November 2004.

[LLT+04] Jason K.S. Lee, Benjamin Lee, Jeremy Thorpe, Kenneth Andrews, Sam Dolinar, and Jon Hamkins. A scalable architecture of a structured LDPC decoder. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Chicago, USA*, page 296, June 2004.

[LMS97] Michael Luby, Michael Mitzenmacher, and Amin Shokrollahi. Practical loss-resilent codes. In *Proc. of 29th ACM Symp. on Theory of Computing (STOC)*, pages 150–159, September 1997.

[LMS98] Michael Luby, Michael Mitzenmacher, and Amin Shokrollahi. Analysis of random process via and-or tree evaluation. In *Proc. of 9th symp. on ACM-SIAM Symposium on Discrete Algorithms*, pages 364–373, 1998.

[LMSS98] Michael Luby, Michael Mitzenmacher, Amin Shokrollahi, and Daniel Spielman. Improved low-density parity-check codes using irregular graphs and belief propagation. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Cambridge, MA, USA*, page 117, August 1998.

[LMSS01a] Michael Luby, Michael Mitzenmacher, Amin Shokrollahi, and Daniel A. Spielman. Efficient erasure correcting codes. *IEEE Trans. Inform. Theory*, 47(2):569–584, February 2001.

[LMSS01b] Michael Luby, Michael Mitzenmacher, Amin Shokrollahi, and Daniel A. Spielman. Improved low-density parity-check codes using irregular graphs. *IEEE Trans. Inform. Theory*, 47(2):585–598, February 2001.

[Lop] LdpcOpt. `http://lthcwww.epfl.ch/research/ldpcopt/`.

[LSL+06] Gianluigi Liva, Shumei Song, Lan Lan, Yifei Zhang, Shu Lin, and William E. Ryan. Design of LDPC codes: A survey and new results. *accepted for publication in J. Comm. Software and Systems*, 2006.

[Lub02] Michael Luby. LT codes. In *Proc. of the 43rd Annual IEEE Symposium on the Foundations of Computer Science (STOC)*, pages 271–280, 2002.

[Mac] MacKay. Encyclopedia of sparse graph codes. `http://www.inference.phy.cam.ac.uk/mackay/codes/data.html`.

[Mac99] David. J. C. MacKay. Good error correcting codes based on very sparse matrices. *IEEE Trans. Inform. Theory*, 45(2):399–431, March 1999.

[May02] Petar Maymounkov. Online codes. Technical report, NYU TR2002-883, November 2002.

[MDG04] Valerian Mannoni, David Declercq, and Guillaume Gelle. Optimized irregular low-density parity-check codes for multicarrier modulations over frequency selective channels. *EURASIP Journ. on Applied Sig. Proc., special issue on 'Multicarrier Communication Systems*, 10:1546–1556, 2004.

[Mit98] Michael Mitzenmacher. A note on low-density parity-check codes for erasures and errors. *SRC Technical Note 1998-017*, December 1998.

[MM03] Petar Maymounkov and David Mazières. Rateless codes and big downloads. In *Proc. of the 2nd Int. Workshop Peer-to-Peer Systems (IPTPS)*, February 2003.

[MMU04] Cyril Measson, Andrea Montanari, and Rudiger Urbanke. Maxwell's construction: the hidden bridge between maximum-likelihood and iterative decoding. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Chicago, USA*, page 225, June 2004.

[MP03] David MacKay and M Post. Weakness of margulis and ramanujan-margulis low-density parity-check codes. In *Proc of MFCSIT2002, Galway*, volume 74 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2003.

[MS06a] Elitza N. Maneva and Amin Shokrollahi. New model for rigorous analysis of LT codes. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Seattle, USA*, July 2006.

[MS06b] Soheil Mohajer and Amin Shokrollahi. Raptor codes with fast hard decision decoding algorithms. In *Proc. of IEEE ITW, Chengdu*, pages 56–60, October 2006.

[Oka03] Toshihiko Okamura. Designing LDPC odes using cyclic shifts. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Yokohama, Japan*, page 151, June 2003.

[OS01] Peter Oswald and Amin Shokrollahi. Capacity-achieving sequences for the erasure channel. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Washington, DC, USA*, page 48, 2001.

[OS02] Peter Oswald and Amin Shokrollahi. Capacity-achieving sequences for the erasure channel. *IEEE Trans. Inform. Theory*, 48(12):3017–3028, December 2002.

[PFD07] Charly Poulliat, Marc Fossorier, and David Declercq. Design of regular $(2, d_c)$ LDPC codes over GF(q) using their binary image. *accepted for publication in IEEE Trans. Commun.*, 2007.

[Pia07] Pablo Piantanida. *Multi-user Information Theory: State Information and Imperfect Channel Knowledge*. PhD thesis, Univ. Paris-Sud XI (Orsay), 2007.

[PMD] Pablo Piantanida, G Matz, and Pierre Duhamel. Outage behavior of discrete memoryless channels under channel estimation errorsl. *submitted to IEEE Trans. on Inform. Theory*, 2006.

[PNF06] Hossein Pishro-Nik and Faramarz Fekri. On raptor codes. In *Proc. of International Conference on Communications (ICC), Istanbul, Turkey*, pages 1137–1141, June 2006.

[Pou04] Charly Poulliat. *Allocation et optimisation de ressources pour la transmission de données multimédia*. PhD thesis, Univ. Cergy-Pontoise, 2004.

[PS06] Payman Pakzad and Amin Shokrollahi. Design principles for raptor codes. In *Proc. of IEEE ITW, Punta del Este, Urugway*, pages 13–20, March 2006.

[PSD07] Pablo Piantanida, Sajad Sadough, and Pierre Duhamel. On the outage capacity of a practical decoder using channel estimation accuracy. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Nice, France*, June 2007.

[PW72] Wesley Peterson and E.J. Weldon. *Error Correcting Codes*. Cambridge, MIT Press, 2nd edition, 1972.

[PY04] Ravi Palanki and Jonathan. S. Yedidia. Rateless codes on noisy channels. In *Proc. of the Conf. on Information Sciences and Systems*, 2004.

[RD07] Aline Roumy and David Declercq. Optimization of ldpc codes for the 2-users gaussian multiple access channel. *EURASIP Journ. on Wireless Communications and Networking*, 2007(Article ID 74890):10 pages, 2007.

[RGCV04] Aline Roumy, Souad Guemghar, Guiseppe Caire, and Sergio Verdú. Design methods for irregular repeat accumulate codes. *IEEE Trans. Inform. Theory*, 50:1711–1727, August 2004.

[Ric03] Tom Richardson. Error floors of LDPC codes. In *Proc. of 41st Allerton Conf*, September 2003.

[RSU01] Thomas J. Richardson, Amin Shokrollahi, and Rudiger Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. Inform. Theory*, 47(2):619–637, February 2001.

[RU01] Thomas J. Richardson and Rudiger Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, 47(2):599–618, February 2001.

[RU04] Tom Richardson and Rudiger Urbanke. Multi-edge type LDPC codes. submitted to IEEE Trans. Inform. Theory, April 2004.

[RU07] Tom Richardson and Rudiger Urbanke. *Modern Coding Theory*. Cambridge University Press, 2007.

[SCRV06] Sundararajan Sankaranarayanan, Shashi Kiran Chilappagari, Rathnakumar Radhakrishnan, and Bane Vasić. Failures of the gallager b decoder: Analysis and applications. In *Proc. of UCSD Workshop on Information Theory and Its Applications*, February 2006.

[SCV04] Stefania Sesia, Giuseppe Caire, and Guillaume . Vivier. Incremental redundancy hybrid ARQ schemes based on low-density parity check codes. *IEEE Trans. Commun.*, 52(8):1311–1321, August 2004.

[Sha48] Claude Elwood Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27, 1948.

[Sha59] Claude Elwood Shannon. Probability of error for optimal codes in a Gaussian channel. *Bell System Technical Journal*, 38:611–656, May 1959.

[Sho06] Amin Shokrollahi. Raptor codes. *IEEE Trans. Inform. Theory*, 52(6):2551–2567, June 2006.

[SVW05] Emina Soljanin, Nedeljko Varnica, and Philip Whiting. Incremental redundancy hybrid ARQ with LDPC and raptor codes. submitted to IEEE Trans. Inform. Theory, September 2005.

[SVW06] Emina Soljanin, Nedeljko Varnica, and Philip Whiting. Punctured vs rateless codes for hybrid ARQ. In *Proc. of IEEE ITW, Punta del Este, Urugway*, pages 155–159, March 2006.

[TAD04] Jeremy Thorpe, Kenneth Andrews, and Sam Dolinar. Methodologies for designing LDPC codes using protographs and circulants. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT), Chicago, USA*, page 236, June 2004.

[Tan81] Michael Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inform. Theory*, 27:533–547, 1981.

[Tho03] Jeremy Thorpe. Low-density parity-check (LDPC) codes constructed from protographs. Technical report, JPL Interplanetary Network Progress (INP) Report 42-154, August 2003.

[TJVW04a] Tao Tian, Christopher R. Jones, John D. Villasenor, and Richard D. Wesel. Construction of irregular LDPC codes with low error floors. In *Proc. of IEEE International Conference on Communications (ICC), Anchorage, Alaska*, pages 3125 – 3129, May 2004.

[TJVW04b] Tao Tian, Christopher R. Jones, John D. Villasenor, and Richard D. Wesel. Selective avoidance of cycles in irregular LDPC code construction. *IEEE Trans. Commun.*, 52(8):1242–1247, August 2004.

[TSF01] Michael R. Tanner, Deepak Sridhara, and Tom Fuja. A class of group-structured LDPC codes. In *Proc. of ISTA*, 2001.

[TV05] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.

[V7.07] 3GPP TS 26.346 V7.6.0. Technical specification group services and system aspects, December 2007.

[Vit37] Andrew J. Viterbi. Error bounds for convolutionnal codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theory*, 13(2):260–269, April 1937.

[Wif06] IEEE 802.11n Wifi. Wireless lan medium access control and physical layer specifications: Enhancements for higher throughput, March 2006.

[WiM05] IEEE 802.16e WiMax. Air interface for fixed and mobile broadband wireless access systems, October 2005.

[XB04] Hua Xiao and Amir H. Banihashemi. Improved progressive-edge-growth (PEG) construction of irregular LDPC codes. *IEEE Commun. Lett.*, 8(12):715–717, December 2004.

[YCLX08] Wending Yao, Lijia Chen, Hui Li, and Hongguang Xu. Research on fountain codes in deep space communication. In *Congress on Image and Signal Processing*, pages 219–224, 2008.