

*Development of Rate-Compatible Structured LDPC CODEC
Algorithms and Hardware IP*

Project Final Report

**School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2006**

Jaehong Kim
Demijan Klinc
Woonhaing Hur
Dr. Aditya Ramamoorthy
Dr. Sunghwan Kim
Dr. Steven W. McLaughlin

Table of Contents

List of Abbreviations

CHAPTER I : Introduction

CHAPTER II : Background Research

 2.1 Low-Density Parity-Check Codes

 2.2 Iterative Decoding Algorithm

 2.3 Efficient Encoding Method

 2.4 Irregular Repeat Accumulate Codes

 2.5 Rate-Compatible Puncturing Algorithm

CHAPTER III : Efficiently-Encodable F

 3.1 New Class of Irregular LDPC Codes

 3.2 Low-Rate Code Design

 3.3 Efficient Encoder Implementation

 3.4 Simulation Results

 3.5 Conclusions

CHAPTER IV : Rate-Compatible L

Redundancy Hybrid ARQ Systems

 4.1 Incremental Redundancy Hybrid ARQ

 4.2 System Model

 4.3 Simulation Results

 4.4 Conclusions

CHAPTER V : Wire-tap Channel Application..... 84

 5.1 Encoding Algorithm for the Wire-tap Channel 84

CHAPTER VI : Optimization of Degree Distributions 92

References..... 96

List of Abbreviations

ACK	Acknowledgment
APP	A Posteriori Probability
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BEC	Binary Erasure Channel
BER	Bit Error Rate
BP	Belief Propagation
BPSK	Binary Phase Shift Keying
BSC	Binary Symmetric Channel
CRC	Cyclic Redundancy Check
CSI	Channel State Information
E ² RC	Efficiently-Iterative Rate-Coded
eIRA	extended IRA
FEC	Forward Error Correction
FER	Frame Error Rate
HARQ	Hybrid ARQ
IC	Integrated Channel
IR	Incremental Redundancy
IRA	Irregular Repeat Accumulation
LDPC	Low-Density Parity Check
LLR	Log Likelihood Ratio
MAP	Maximum A Posteriori
MMSE	Minimum Mean Square Error
NACK	Negative Acknowledgment
PEG	Progressive Edge-Growth
QC	Quasi-Cyclic
QPSK	Quadrature Phase Shift Keying
RCPC	Rate-Compatible Punctured
V-BLAST	Vertical Bell Labs Layered Space-Time
VLSI	Very Large Scale Integration
WER	Word Error Rate

CHAPTER I

INTRODUCTION

Low-density parity-check (LDPC) codes by Gallager [1] had been forgotten for several decades in spite of their excellent properties, since the implementation of these codes seemed to be impossible at that time. These codes were rediscovered in the middle of the 1990s [2] and were shown to achieve Shannon limit within 0.0045dB [3]. LDPC codes are now considered good candidates for the next-generation forward error correction (FEC) technique in high throughput wireless and recording applications. Their excellent performance and iterative decoder make them appropriate for technologies such as DVB-S2, IEEE 802.16e [4], and IEEE 802.11n [5], [6].

While semiconductor technology has progressed to an extent where the implementation of LDPC codes has become possible, many practical issues still remain. First and foremost, there is a need to reduce complexity without sacrificing performance. Second, for applications such as wireless LANs, the system throughput depends upon the channel conditions and hence the code needs to have the ability to operate at different rates. Third, while the LDPC decoder can operate in linear time, it may be hard to perform low-complexity encoding of these codes. In particular, the class of irregular LDPC codes introduced by Richardson *et al.* [7] may have high memory and processing requirements, especially at short block lengths. While the encoding time can be reduced substantially using the techniques presented in [8] at long block lengths, their techniques may be hard to apply at short block lengths. The other option is to resort quasi-cyclic

(QC) LDPC or algebraic constructions that can be encoded in linear time.

Irregular repeat-accumulate (IRA) codes were introduced in [9]. These codes have a linear-time encoder and their performance is close to that of LDPC codes. This class of codes was extended, called RCPC codes, by Yang *et al.* [11], where they demonstrated high-rate codes.

A popular technique for achieving rate adaptation is rate-compatible puncturing (RCPC). A rate-compatible puncturing scheme is applying to incremental redundancy (IR) hybrid ARQ (HARQ) systems, since the parity bit set of a higher rate code can be used as a lower rate code [12]. The RCPC scheme has another advantage: the same encoder and decoder while operating at different rates. The transmitter sends depends on the rate requirement. The bits that are not transmitted are treated as erasures. Thus, puncturing is a good solution to the rate-adaptation problem.

Motivated by these observations, this report first introduces a class of LDPC codes with short block lengths. Based on the observations, a class of codes is proposed that can be efficiently encoded and decoded in a rate-compatible fashion. The proposed LDPC codes have a linear-time encoder and have good performance under puncturing. In the next section, we verify that the proposed codes show good throughput when applied to IR-HARQ systems over time-varying channels.

CHAPTER II

BACKGROUND RESEARCH

Channel coding is an essential technique to cope with errors occurring in channels of communication systems and storage systems. Channel coding has flourished in two branches. Channel errors can be corrected with forward error correction (FEC) codes. On the other hand, a receiver may request retransmission of the previous data if it fails to recover them, which is called automatic repeat request (ARQ). FEC codes can be classified into block codes, such as cyclic codes and LDPC codes, and tree codes, such as convolutional codes and Turbo codes. In this chapter, we briefly explain the block codes where LDPC codes are specified.

Let us consider linear block codes over the binary field $F_2 = \{0, 1\}$ with $+$, \times . Let F_2^N be the N -dimensional vector space over F_2 . Then, an (N, K) linear block code C is defined as K -dimensional subspace of F_2^N , where K is a data word length and N is a codeword length. Since C is a subspace of dimension K , there are K linearly independent vectors g_0, g_1, \dots, g_{K-1} which span C . Let $m = [m_0, m_1, \dots, m_{K-1}]$ be the data word and $c = [c_0, c_1, \dots, c_{N-1}]$ be the corresponding codeword in the code C . The mapping $m \rightarrow c$ is thus naturally written as $c = m_0 g_0 + m_1 g_1 + \dots + m_{K-1} g_{K-1}$. This relationship can be represented in the matrix form $c = mG$, where G is a $K \times N$ matrix;

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{K-1} \end{bmatrix}$$

We call the matrix G the generator matrix for C . The encoding process can be viewed as an injective mapping from the K -dimensional vector space into vectors from the N -dimensional vector space. The ratio

$$R = \frac{K}{N}$$

is called *code rate*.

On the other hand, the null space C^\perp of C has $N - K$ linearly independent vectors $h_0, h_1, \dots, h_{N-K-1}$. These vectors have for any $c \in C$ that

$$h_i \cdot c^T = 0, \quad \forall i$$

This relationship can be represented in the matrix form. Let H be the so-called *parity-check matrix* defined as

$$H = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{N-K-1} \end{bmatrix}$$

A low-density parity-check code is so called because of its low density of 1s. We address the details of LDPC codes in the next chapter.

2.1 LOW-DENSITY PARITY-CHECK CODES

Every LDPC code is uniquely specified by its parity-check matrix H or, equivalently, by means of the *Tanner graph* [13], as illustrated in Figure 2.1. The Tanner graph consists of two types of nodes: *variable nodes* and *check nodes*, which are connected by edges. Since there can be no direct connection between any two nodes of the same type, the Tanner graph is said to be *bipartite*. Consider an LDPC code defined by its corresponding Tanner graph. Each variable node, depicted by a circle, represents one bit of a codeword, and every check node, depicted by a square, represents one parity-check equation.

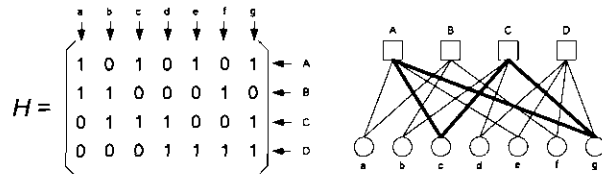


Figure 2.1 A parity-check matrix and its Tanner graph; Thick lines in the graph implies cycle 4.

Since we are considering N codeword length and K data word length, the Tanner graph contains N variable nodes and M check nodes, where $M = N - K$. Let us denote the parity-check matrix $H = (h_{ij})_{1 \leq i \leq M, 1 \leq j \leq N}$. Then, the i -th check node is connected to the j -th variable node if and only if $h_{ij} = 1$. For example, 1 in column f and row D in the parity-check matrix in Figure 2.1 corresponds to an edge connection between variable

node f and check node D in the Tanner graph. If the variable or check node, we say that node has degree 2 and check node D has degree 4. Tanner graph visualization tool for a variety of issues concerning

Definition 2.1: A cycle of length l in a Tanner graph begins and ends at the same node, whereby every

The length of a cycle is the number of edges in the cycle. Many cycles of different lengths in their Tanner graph

Definition 2.2: The girth in a Tanner graph is the length of the shortest cycle.

The girth has a great importance for the code's performance. In a bipartite graph, the smallest girth has length 4, as shown in Figure 2.1. However, it is desirable to avoid short cycles in design because they can cause poor performance.

An ensemble of LDPC codes is defined by two degree distributions, called a *degree distribution pair*, for the variable and check nodes.

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1}$$

$$\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1}$$

where λ_i is the fraction of edges emanating from variable nodes of degree i .

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.