

中图分类号:

UDC:

学校代码: 10055

密级: 公开

南开大学  
硕士学位论文

GF(q) 上 LDPC 分组码和卷积码的比较

Comparison of LDPC Block and Convolutional Codes over GF(q)

论文作者 \_\_\_\_\_ 指导教师 \_\_\_\_\_

申请学位 \_\_\_\_\_ 培养单位 \_\_\_\_\_

学科专业 \_\_\_\_\_ 研究方向 \_\_\_\_\_

答辩委员会主席 \_\_\_\_\_ 评阅人 \_\_\_\_\_

南开大学研究生院

二〇一六年五月

## 南开大学学位论文使用授权书

根据《南开大学关于研究生学位论文收藏和利用管理办法》，我校的博士、硕士学位获得者均须向南开大学提交本人的学位论文纸质本及相应电子版。

本人完全了解南开大学有关研究生学位论文收藏和利用的管理规定。南开大学拥有在《著作权法》规定范围内的学位论文使用权，即：(1) 学位获得者必须按规定提交学位论文(包括纸质印刷本及电子版)，学校可以采用影印、缩印或其他复制手段保存研究生学位论文，并编入《南开大学博硕士学位论文全文数据库》；(2) 为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆等场所提供校内师生阅读，在校园网上提供论文目录检索、文摘以及论文全文浏览、下载等免费信息服务；(3) 根据教育部有关规定，南开大学向教育部指定单位提交公开的学位论文；(4) 学位论文作者授权学校向中国科技信息研究所和中国学术期刊(光盘) 电子出版社提交规定范围的学位论文及其电子版并收入相应学位论文数据库，通过其相关网站对外进行信息服务。同时本人保留在其他媒体发表论文的权利。

非公开学位论文，保密期限内不向外提交和提供服务，解密后提交和服务同公开论文。

论文电子版提交至校图书馆网站：<http://202.113.20.161:8001/index.htm>。

本人承诺：本人的学位论文是在南开大学学习期间创作完成的作品，并已通过论文答辩；提交的学位论文电子版与纸质本论文的内容一致，如因不同造成不良后果由本人自负。

本人同意遵守上述规定。本授权书签署一式两份，由研究生院和图书馆留存。

作者暨授权人签字：\_\_\_\_\_

20     年     月     日

### 南开大学研究生学位论文作者信息

论 文 题 目	GF(q) 上 LDPC 分组码和卷积码的比较				
姓 名		学号		答辩日期	
论 文 类 别	博士 <input type="checkbox"/> 学历硕士 <input type="checkbox"/> 硕士专业学位 <input type="checkbox"/> 高校教师 <input type="checkbox"/> 同等学力硕士 <input type="checkbox"/>				
院 / 系 / 所			专 业		
联 系 电 话			Email		
通讯地址 (邮编):					
备注:					

注:本授权书适用我校授予的所有博士、硕士的学位论文。由作者填写 (一式两份) 签字后交校图书馆，非公开学位论文须附《南开大学研究生申请非公开学位论文审批表》。

# 南开大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下进行研究工作所取得的研究成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：\_\_\_\_\_年 月 日

## 非公开学位论文标注说明

根据南开大学有关规定，非公开学位论文须经指导教师同意、作者本人申请和相关部门批准方能标注。未经批准的均为公开学位论文，公开学位论文本说明为空白。

论文题目			
申请密级	<input type="checkbox"/> 限制 (≤2 年) <input type="checkbox"/> 秘密 (≤10 年) <input type="checkbox"/> 机密 (≤20 年)		
保密期限	20 年 月 日至 20 年 月 日		
审批表编号		批准日期	20 年 月 日

南开大学学位办公室盖章 (有效)

限制 ☐ 2 年（最长 2 年，可少于 2 年）

秘密 ☐ 10 年（最长 5 年，可少于 5 年）

机密 ☐ 20 年（最长 10 年，可少于 10 年）

## 摘 要

本文介绍了  $GF(q)$  上的 LDPC 分组码, 并从 LDPC 分组码构建结构相似的 LDPC 卷积码。接着分析了 LDPC 分组码的和积算法, 以及用于 LDPC 卷积码的滑动窗口和积算法。在不同参数的  $GF(q)$  上, 本文主要考虑了两种情况: 一种是 LDPC 分组码解码延时与 LDPC 卷积码的解码延时相等的情况; 另一种是 LDPC 分组码码长与 LDPC 卷积码的限制长度相等的情况。基于以上条件, 本文比较了多元 LDPC 分组码与二元、多元 LDPC 卷积码的解码性能, 其中模拟的结果表明  $q$  元 LDPC 卷积码的解码性能好于 2 元 LDPC 卷积码, 以及  $q$  元 LDPC 卷积码的解码性能好于  $q$  元 LDPC 分组码。最后, 本文还在解码延时相等的条件下, 比较了  $q$  元 LDPC 卷积码与  $q$  元 LDPC 分组码的解码计算复杂度。模拟结果表明, 解码性能与计算复杂度两种条件的平衡中, LDPC 卷积码较 LDPC 分组码表现更优。

**关键词:** LDPC 分组码; LDPC 卷积码; 滑动窗口; 和积算法; 解码延时

## Abstract

In this paper, we introduce LDPC block codes (LDPC-BC), and LDPC convolutional codes (LDPC-CC) which are derived from the former. Then we analyse sum product algorithm (SPA) for LDPC-BC, and a sliding window decoder (WD) for LDPC-CC. Based on different  $GF(q)$ , we compare the decoding performance between  $q$ -ary LDPC-BC and LDPC-CC in two regimes: one when the constraint length of  $q$ -ary SC-LDPC codes is equal to the block length of  $q$ -ary LDPC-BCs and the other when the two decoding latencies are equal. Simulation shows that  $q$ -ary LDPC-CC outperform binary LDPC-CC and  $q$ -ary LDPC-BC. We also compared computational complexity of  $q$ -ary LDPC-CC and  $q$ -ary LDPC-BC under equal decoding latency assumptions. Simulation shows that 4-ary LDPC-CC outperform other LDPC-CCs and LDPC-BCs in trade-off between computational complexity and decoding latency.

**Key Words:** LDPC block codes; LDPC convolutional codes; Sliding window decoder; Belief propagation; Decoding latency

## 目 录

摘要 .....	I
Abstract .....	II
第一章 背景及介绍 .....	1
第一节 背景 .....	1
第二节 本文结构 .....	3
第二章 基于原模图构造的 LDPC 码 .....	4
第一节 LDPC 码的原模图及 LDPC 码的叠加方式 .....	4
第二节 LDPC 分组码的构造 .....	6
第三节 LDPC 卷积码的构造 .....	7
第三章 和积算法与滑动窗口解码的简介 .....	11
第一节 LDPC 分组码的和积算法 .....	11
第二节 LDPC 卷积码的滑动窗口解码 .....	12
第四章 模拟仿真结果 .....	14
第一节 LDPC 分组码码长与 LDPC 卷积码限制长度相等 .....	14
第二节 LDPC 分组码与 LDPC 卷积码的解码延时相等 .....	15
第三节 LDPC 分组码与 LDPC 卷积码的解码复杂度比较 .....	18
第五章 总结 .....	21
参考文献 .....	22

## 第一章 背景及介绍

### 第一节 背景

低密度校验分组码 (LDPC-BC), 当其结合使用复杂度低的置信传播解码算法时, 是一类接近香农极限的编码。另外, 它还有另一个优点, 即其解码复杂度随着分组码码长增加而线性增加。Gallager 在 1962 年提出低密度校验分组码 [1]。在同一篇文章中, 他还提出了一种信息遍历解码算法。由于当时计算机计算能力不高, LDPC 码并未得到重视, 直到九十年代人们才重新认识到 LDPC 码的优点。在 2001 年 Richardson 等人的模拟试验中 [2], 用信息遍历算法解码 LDPC 分组码的性能超过了 turbo 码, 并且接近香农极限。而 Davey 和 MacKay 考虑了定义在有限域  $GF(q)$  的 LDPC 分组码 [3], 同时把 Gallager 的置信传播解码算法推广到  $q$  元域, 称作  $q$  元和积算法 (QSPA)。该文章中的实验证明  $q$  元 LDPC 分组码具有很好的性能。2003 年, Barnault 等人将快速傅立叶变换引入和积算法中, 将和积算法推广为 FFT-QSPA, 使得和积算法的复杂度有所降低 [4]。另外, 2004 年 Wymeersch 在和积算法引入对数似然比的概念, 避免了实数乘法运算 [5]。之后, 拓展最小和算法 (EMS) 被提出, 它能进一步降低解码的复杂度。由于码长中等和码长较短的 LDPC 分组码有极好的解码性能,  $q$  元 LDPC 分组码最近获得学术界极大关注。

与 LDPC 分组码相对应的卷积码, 称为 LDPC 卷积码, 由 Felström 和 Zigangirov[6] 于 1999 年提出。LDPC 卷积码与 LDPC 分组码类似, 能通过稀疏校验矩阵定义, 所以 LDPC 卷积码解码也可以用信息遍历解码算法。LDPC 卷积码的特点是, 如果该 LDPC 卷积码是从某一类正则或非正则 LDPC 分组码衍生构造的话, LDPC 卷积码的置信传播解码算法阈值能渐近达到, 它所基于的 LDPC 分组码的最大后验概率 (MAP) 解码阈值 [7]。随后, 某一类从正则 LDPC 分组码衍生构造出来的 LDPC 卷积码被证明存在阈值饱和现象, 即 LDPC 卷积码如果在无记忆对称二元信道中使用置信传播解码算法的话, 能够达到它所基于的 LDPC 分组码的 MAP 阈值, 也就是说正则 LDPC 卷积码能够通过增加校验矩阵的密度达到信道容量。

由于 LDPC 卷积码与 LDPC 分组码具有相似的结构, Costello 在 [8] 中比较了二元 LDPC 分组码与二元 LDPC 卷积码的性能。在这篇文章中, Costello 使用了一种并行高速流水线解码算法, 这种算法的具体实现方式在 Pusane 等人的 [9] 中有深入的介绍。但是为了达到接近香农极限的性能, 该算法必须经过多次迭代, 而解码延时和内存占用量与算法迭代次数正相关, 所以, 要达到香农极限, 其内存占用量与译码延时大大影响了解码效率。为了解决 LDPC 卷积码使用流水线解码算法时的缺点, Iyengar 等人提出了置信传播解码算法的滑动窗口版本 [10], 并研究了算法中解码延时和解码性能之间的取舍。在 [11] 中, Hassan 等人将改进后的滑动窗口解码算法应用于迭代解码阈值分析。在 [12] 中, Hehn 等人基于结构延时, 对 LDPC 分组码与 LDPC 卷积码进行了比较, 其中 LDPC 卷积码使用滑动窗口解码算法进行解码。

在构建 LDPC 卷积码方面, Felström 和 Zigangirov 做出了开创性工作 [6]。并且, 在这篇文章中, Felström 和 Zigangirov 证明了在二元擦除信道中,  $q$  元 LDPC 卷积码也具有二元 LDPC 卷积码阈值饱和现象。随后基于原模图 [13] 构造 LDPC 卷积码的方法被提出。结合原模图与边逐步增长方法 (PEG)[14], LDPC 卷积码能很方便地构造出来。最近, [15] 中分析了由原模图构造的  $q$  元 LDPC 卷积码在滑动窗口解码算法中的解码阈值表现。因此这篇文章也为设计与实现适用于滑动窗口解码算法的  $q$  元 LDPC 卷积码提供了理论指导。

相对于 Hehn 等人的 [12] 所考虑的二元 LDPC 卷积码的解码性能不同, 本文主要考虑有限长度的  $q$  元 LDPC 卷积码以及  $q$  元 LDPC 分组码, 模拟仿真所基于的信道是二元对称信道。由于流水线解码算法会造成较大的解码延时, 本文在解码  $q$  元 LDPC 卷积码时主要采用滑动窗口解码算法。另外, 为了减小计算的复杂度, 本文还引入了一个基于置信传播软判决的迭代停止规则。在比较  $q$  元 LDPC 卷积码与  $q$  元 LDPC 分组码的性能时, 本文主要考虑两种情况: 一种情况是  $q$  元 LDPC 卷积码的限制长度等于  $q$  元 LDPC 分组码的码长; 另一种情况是  $q$  元 LDPC 卷积码的解码延时与  $q$  元 LDPC 分组码的解码延时相等。本文在  $q$  元域上的比较方法与 [12] 中二元域的比较方法类似, 但是本文也有自己的重点。比如, 对于由 (2,4) 正则 LDPC 码构造的  $q$  元 LDPC 分组码与  $q$  元 LDPC 卷积码之间的比较, 我们将码长参数限制在相对小的区间。本文还考察了, 当解码延时确定时, 原模图复制数, 解码窗口大小与  $q$  元 LDPC 卷积码解码性能之间的关系。最后, 本文还在解码性能或者解码延时确定的条件下, 比较  $q$  元



LDPC 分组码与  $q$  元 LDPC 卷积码之间解码算法的计算复杂度。

## 第二节 本文结构

本文的结构如下：第一章主要介绍 LDPC 卷积码与 LDPC 分组码的历史及背景。第二章主要描述 LDPC 卷积码与 LDPC 分组码的构建过程，同时指出公平比较两者性能所基于的构建规则。第三章主要描述 LDPC 卷积码与 LDPC 分组码的解码算法，并且计算了两种算法的解码延时。在第四章，通过调整几类参数，我们比较了 LDPC 卷积码与 LDPC 分组码的性能。第五章是结论及总结以上内容。

## 第二章 基于原模图构造的 LDPC 码

### 第一节 LDPC 码的原模图及 LDPC 码的叠加方式

许多好的 LDPC 码都是通过人工叠加短 LDPC 码而形成的。LDPC 码的叠加能很形象地通过 Tanner 图的方式表现出来。考虑一个短 LDPC 码，以及与其对应的小的 Tanner 图，如图2.1。这个小的 Tanner 图有多个名字，如原模图、基本图或者投影图。从原模图、基本图这两个名字可以看出，这种小的 Tanner 图是组成更大的 Tanner 图的基础。而投影图这个名字则说明了，这个小的 Tanner 图是叠加其自身而构成更大的 Tanner 图的投影。图2.1中圈中等号代表变量节点，圈中加号代表校验节点，该图右侧为对应 LDPC 码的校验矩阵。

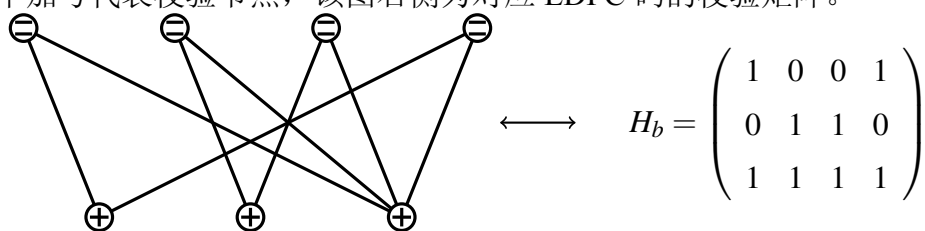


图 2.1 基本图

对于规则的 LDPC 码，可以采取更简单的原模图来描述。由于规则的 LDPC 码的每一个信息节点与校验节点都有相同的度分布，即相同的边连接模式，所以我们可以将基本图折叠起来，让它变成如图2.2的形状。图2.2表示的是 (3,6) 正则 LDPC 码的原模图以及其对应的基本矩阵描述。

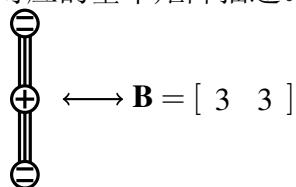
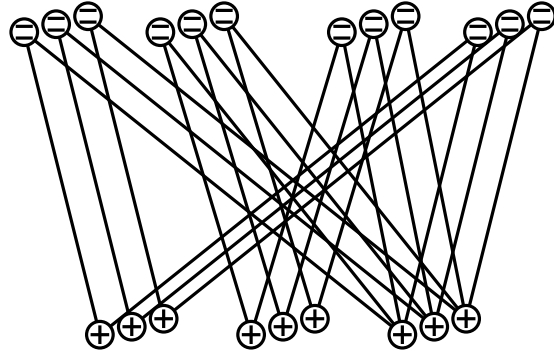


图 2.2 LDPC 码的原模图

为了构造有结构的 LDPC 码，首先要将图2.1复制  $m$  次。此时变量节点、校验节点以及节点间相连的边都变成了  $m$  份，形成了一个簇状的 Tanner 图，如图2.3。

图 2.3 复制  $m$  次基本图

原模图被复制了  $m$  次之后，虽然变量节点、校验节点以及节点间相连的边都变成了  $m$  份，但是此时这  $m$  个原模图还是隔离的。所以要对每一个边的簇应用置换方法，使得不同的原模图之间相连起来。此时便完成了从原模图构造如原模图  $m$  倍大小的有结构的 LDPC 码，如图 2.4。

如果采用矩阵的表述方式，一开始我们有一个 LDPC 短码的校验矩阵  $H_b$ ，如图 2.1 的右边的矩阵所示。令  $\mathcal{P}$  为  $m \times m$  维置换矩阵的集合。我们通过把  $H_b$  中的每个元素换成  $m \times m$  的矩阵来形成  $m$  倍大的 LDPC 码。其中，把  $H_b$  中为 1 的元素换成  $\mathcal{P}$  中的某个矩阵，把  $H_b$  中为 0 的元素换成零矩阵。通过这种复制方式，LDPC 短码的校验矩阵  $H_b$  变成了  $m$  倍大的校验矩阵：

$$H = \begin{pmatrix} P^2 & 0 & 0 & P^1 \\ 0 & P^3 & P^1 & 0 \\ P^1 & P^3 & P^2 & P^2 \end{pmatrix} \quad (2.1)$$

其中  $P^i$  对应循环置换矩阵，其意义是对集合  $\{1, \dots, m\}$  进行移位操作  $i-1$  次：它的列向量对应于变量节点，行向量对应于校验节点。比如  $P^1$  对应单位矩阵，以及

$$P^2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (2.2)$$

$P^2$  表示的意思是将第一个原模图的第一个变量节点连接到第二个原模图的第一个校验节点，将第二个原模图的第一个变量节点连接到第三个原模图的第一个校验节点，将第三个原模图的第一个变量节点连接到第一个原模图的第一个校验节点。

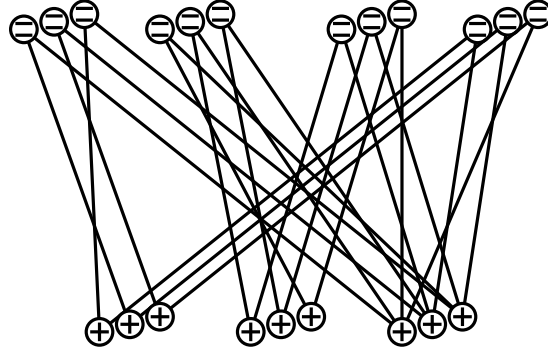


图 2.4 具有一定结构的 LDPC 码

## 第二节 LDPC 分组码的构造

设计码率为  $R = b/c$  的 LDPC 分组码的原模图有  $c$  个变量节点和  $c - b$  个校验节点。它能生成不同码长的，设计码率为  $R$ ，有相同的度分布的分组码。一个有  $c = 2$  个变量节点，变量节点的度为 3，和  $c - b = 1$  个校验节点，校验节点的度为 6 的分组码原模图的例子可以参看图 2.1。

下面将描述  $q$  元 LDPC 分组码的构造方法。 $\text{GF}(q)$  为含  $q = 2^m$  个元素的有限域，其中  $m$  为在  $\text{GF}(q)$  代表一个符号所需的位数。令  $M$  为原模图叠加数，其中  $M$  是一个大整数。通过以下两步从原模图的  $(c - b) \times c$  邻接矩阵  $\mathbf{B} = [B_{i,j}]$  构造码长为  $n_{BC} = Mc$  的  $q$  元 LDPC 分组码：

1. 将  $\mathbf{B}$  中的非零元  $B_{i,j}$  替换为随机选择的  $M \times M$  置换矩阵，将  $\mathbf{B}$  中的零元  $B_{i,j}$  替换为  $M \times M$  零矩阵。此时得到对应于  $\mathbf{B}$  的二元校验矩阵  $\mathbf{H}$ ；
2. 将  $\mathbf{H}$  中非零元替换为从有限域  $\text{GF}(q)$  中随机选取的元素，得到 LDPC 分组码的  $q$  元校验矩阵  $\mathbf{H}_{BC}$ 。

对于 LDPC 分组码，必须等待整个码块接受完毕才能执行置信传播解码算法。故  $q$  元 LDPC 分组码的解码延时为

$$T_{BC} = n_{BC} \cdot m = Mmc \quad (2.3)$$

### 第三节 LDPC 卷积码的构造

对于给定码率为  $R = b/c$  的 LDPC 卷积码，其定义为，存在校验矩阵  $\mathbf{H}_{[\infty]}$  使得无限长向量  $\mathbf{v}_{[\infty]}$  有  $\mathbf{H}_{[\infty]} \mathbf{v}_{[\infty]}^T = \mathbf{0}_{[\infty]}$ ，其中

$$\mathbf{H}_{[\infty]} = \begin{bmatrix} \mathbf{H}_0(1) & & & & \\ \mathbf{H}_1(1) & \mathbf{H}_0(2) & & & \\ \vdots & \mathbf{H}_1(2) & \ddots & & \\ \mathbf{H}_{m_s}(1) & \vdots & \ddots & \mathbf{H}_0(t) & \\ & \mathbf{H}_{m_s}(2) & \ddots & \mathbf{H}_1(t) & \ddots \\ & & \ddots & \vdots & \ddots \\ & & & \mathbf{H}_{m_s}(t) & \ddots \\ & & & & \ddots \end{bmatrix} \quad (2.4)$$

而  $\mathbf{0}_{[\infty]}$  是无限长零向量。 $\mathbf{H}_i(t), i = 0, 1, \dots, m_s$  为  $(c-b) \times c$  的矩阵满足以下条件：

- $\mathbf{H}_i(t) = \mathbf{0}$ ，当  $i < 0$  和  $i > m_s$ ， $\forall t \geq 1$ ；
- $\exists t \geq 0$ ，使得  $\mathbf{H}_{m_s}(t) \neq \mathbf{0}$ ；
- $\forall t \geq 1$ ， $\mathbf{H}_0(1)$  满秩。

其中参数  $m_s$  称为 LDPC 卷积码的记忆因子。 $v = (m_s + 1)c$  为 LDPC 卷积码的限制长度。对于  $\forall t, \tau > 1, \mathbf{H}_i(t) = \mathbf{H}_i(t + \tau), \forall i = 0, 1, \dots, m_s$  的情况，我们称 LDPC 卷积码具有周期性。若  $\tau = 1$ ，称 LDPC 卷积码为时不变的。另外，截尾的 LDPC 卷积码的校验矩阵长度有限，即在  $L$  时刻终止，如

$$\mathbf{H}_{[\infty]} = \begin{bmatrix} \mathbf{H}_0(1) & & & & \\ \mathbf{H}_1(1) & \mathbf{H}_0(2) & & & \\ \vdots & \mathbf{H}_1(2) & \ddots & & \\ \mathbf{H}_{m_s}(1) & \vdots & \ddots & \mathbf{H}_0(L) & \\ & \mathbf{H}_{m_s}(2) & \ddots & \mathbf{H}_1(L) & \\ & & \ddots & \vdots & \\ & & & \mathbf{H}_{m_s}(L) & \end{bmatrix} \quad (2.5)$$

下面通过原模图来构造截尾的时不变的 LDPC 卷积码。设原模图的基本矩

阵为  $(c-b) \times c$  的  $\mathbf{B}$ 。首先构造  $\mathbf{B}_{CC}$

$$\mathbf{B}_{CC} = \begin{bmatrix} \mathbf{B}_0 & & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & & \\ \vdots & \mathbf{B}_1 & \ddots & & \\ \mathbf{B}_{m_s} & \vdots & \ddots & \ddots & \\ & \mathbf{B}_{m_s} & \ddots & \ddots & \ddots \end{bmatrix} \quad (2.6)$$

其中  $m_s$  为在图2.5中当前原模图与前一个原模图相连的边数。 $\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_{m_s}$  为  $(c-b) \times c$  矩阵，且满足

$$\sum_{i=0}^{m_s} \mathbf{B}_i = \mathbf{B} \quad (2.7)$$

然后将  $\mathbf{B}_{CC}$  中的非零元替换为随机选择的  $M \times M$  置换矩阵，将  $\mathbf{B}_{CC}$  中的零元替换为  $M \times M$  零矩阵，得到 LDPC 卷积码校验矩阵  $\mathbf{H}_{CC}$ 。最后将  $\mathbf{H}_{CC}$  中的非零元替换为从有限域  $\text{GF}(q)$  中随机选取的元素，得到 LDPC 卷积码的  $q$  元校验矩阵  $\mathbf{H}_{CC}$ ，其限制长度为  $v_s = (m_s + 1)Mc$ 。

图2.5为通过原模图叠加方法构造 LDPC 卷积码的示意图。首先复制几次 (3,6) 正则 LDPC 码的原模图，设定记忆因子  $m_s = 1$ ，然后将原模图之间连接起来。其中，原模图的基本矩阵为  $\mathbf{B} = \begin{bmatrix} 3 & 3 \end{bmatrix}$ ，使用到的组成矩阵为  $\mathbf{B}_0 = \begin{bmatrix} 2 & 1 \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} 1 & 2 \end{bmatrix}$ 。

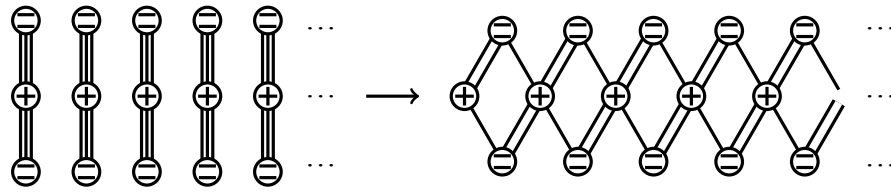


图 2.5 构造 LDPC 卷积码

由于记忆参数  $m_s = 1$  的 LDPC 卷积码有优秀的解码阈值，以及在有限长情况下，使用窗口参数较小的滑动窗口译码算法能达到较好的性能，在之后的实证分析中，本文只考虑记忆参数  $m_s = 1$  的 LDPC 卷积码。并且本文只考虑  $(d_v, d_c)$  正则 LDPC 卷积码，即  $\mathbf{H}_{CC}$  中每行重量为  $d_c$ ，每列重量为  $d_v$ 。这是因为  $(d_v, d_c)$  正则 LDPC 卷积码具有逼近信道容量的性能，以及其解码复杂度低于非正则 LDPC 卷积码。

为了公平地比较 LDPC 卷积码与 LDPC 分组码的性能, 本文限定构造 LDPC 卷积码与 LDPC 分组码的校验矩阵的方法如下。选择两个  $(c-b) \times c$  矩阵  $\mathbf{B}_0$  和  $\mathbf{B}_1$ , 使得  $\mathbf{B}_0 + \mathbf{B}_1$  为  $(d_v, d_c)$  正则矩阵。 $(d_v, d_c)$  正则 LDPC 分组码的基本矩阵为

$$\mathbf{B}_{BC} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 \\ \mathbf{B}_1 & \mathbf{B}_0 \end{bmatrix}_{2(c-b) \times 2c} \quad (2.8)$$

其中  $\mathbf{B}_{BC}$  每个列向量的重量为  $d_v$ , 每个行向量的重量为  $d_c$ 。然后使用第二节的原模图叠加方法构造 LDPC 分组码的校验矩阵

$$\mathbf{H}_{BC} = \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ \mathbf{H}_1 & \mathbf{H}_0 \end{bmatrix}_{2(c-b)M \times 2cM} \quad (2.9)$$

对于 LDPC 卷积码, 使用相同的  $\mathbf{B}_0$  和  $\mathbf{B}_1$ , 得到  $(d_v, d_c)$  正则 LDPC 卷积码的基本矩阵为

$$\mathbf{B}_{CC} = \begin{bmatrix} \mathbf{B}_0 & & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & & \\ & \mathbf{B}_1 & \mathbf{B}_0 & & \\ & & \mathbf{B}_1 & \mathbf{B}_0 & \\ & & & \mathbf{B}_1 & \ddots \\ & & & & \ddots \end{bmatrix} \quad (2.10)$$

然后使用原模图叠加方法构造 LDPC 卷积码的校验矩阵

$$\mathbf{H}_{CC} = \begin{bmatrix} \mathbf{H}_0 & & & & \\ \mathbf{H}_1 & \mathbf{H}_0 & & & \\ & \mathbf{H}_1 & \mathbf{H}_0 & & \\ & & \mathbf{H}_1 & \mathbf{H}_0 & \\ & & & \mathbf{H}_1 & \ddots \\ & & & & \ddots \end{bmatrix} \quad (2.11)$$

基于以上构造规则, 可以使用其他的基本矩阵 (见表2.1) 构造 LDPC 分组码校验矩阵  $\mathbf{H}_{BC}$  与 LDPC 卷积码校验矩阵  $\mathbf{H}_{CC}$ 。当原模图叠加数为  $M$ , 有限域元素个数为  $q = 2^m$  时, LDPC 分组码的码长和 LDPC 卷积码的限制长度都等于  $2Mmc$ 。

从 LDPC 卷积码的校验矩阵  $\mathbf{H}_{CC}$  以及 LDPC 分组码的校验矩阵  $\mathbf{H}_{BC}$  可以看出,  $\mathbf{H}_{CC}$  中的某一部分在周期性地重复  $\mathbf{H}_{BC}$ 。这种构造方式源于 [6] 中, 从 LDPC 分组码衍生 LDPC 卷积码的展开算法。同时我们还可以注意到, 尽管我们称由  $\mathbf{H}_{CC}$  生成的 LDPC 卷积码是  $(d_v, d_c)$  正则的, 但是  $\mathbf{H}_{CC}$  并非真正的  $(d_v, d_c)$  正则校验矩阵, 因为它的前  $(c-b)$  行的重量小于  $d_c$ 。而正是因为  $(d_v, d_c)$  正则 LDPC 卷积码具有这种微小的结构不规则性, 其解码阈值才能逼近信道容量。

表 2.1 构造 LDPC 分组码和 LDPC 卷积码的组成矩阵

码	组成矩阵	码长/限制长度
(2,4) 正则	$\mathbf{B}_0 = \mathbf{B}_1 = \begin{bmatrix} 1 & 1 \end{bmatrix}$	4Mm
(3,6) 正则	$\mathbf{B}_0 = \begin{bmatrix} 2 & 1 \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} 1 & 2 \end{bmatrix}$	4Mm



## 第三章 和积算法与滑动窗口解码的简介

### 第一节 LDPC 分组码的和积算法

和积算法是一种软判决算法。在算法迭代过程中，校验节点生成独立于信息节点接收到信息的额外信息，进而决定信息节点的值。以下介绍基本的和积算法。

记信源发送码字为  $\mathbf{c}$ ，接收到的向量为  $\mathbf{y}$ 。将从校验节点  $j$  到它所连接信息节点  $i$  额外信息记为  $E_{j,i}$ 。如果某次迭代中，码字中  $\mathbf{c}_{i'} = 1$  的概率为  $P_{j,i'}$ ，那么校验方程中包含奇数个 1 的概率为

$$P_{j,i}^{ext} = \frac{1}{2} - \frac{1}{2} \prod_{i' \in B_j, i' \neq i} (1 - 2P_{j,i'}) \quad (3.1)$$

其中， $B_j$  为与校验节点  $j$  相连的信息节点的下标集合。类似的，校验方程满足  $\mathbf{c}_i = 0$  时概率为  $1 - P_{j,i}^{ext}$ 。

每个信息节点接收输入的 LLR，记为  $R_i$ ，若信道为对称信道，即  $p(\mathbf{c}_i = 0) = p(\mathbf{c}_i = 1)$

$$R_i = \log \frac{p(\mathbf{c}_i = 0|y_i)}{p(\mathbf{c}_i = 1|y_i)} = \log \frac{p(y_i|\mathbf{c}_i = 0)}{p(y_i|\mathbf{c}_i = 1)} \quad (3.2)$$

对于不同的信道模型以及调制方式， $R_i$  的具体形式都是不同的。对于本文使用翻转概率为  $p$  的 BSC 信道而言

$$R_i = \begin{cases} \frac{1-p}{p}, & y_i = 0 \\ \frac{p}{1-p}, & y_i = 1 \end{cases} \quad (3.3)$$

对于 AWGN 信道, BPSK 调制方式而言, 调制后的信号幅度为  $s = \{\sqrt{E_b}, -\sqrt{E_b}\}$ , 对应  $\mathbf{c}_i = 0$  和  $\mathbf{c}_i = 1$ , 则

$$R_i = \log \frac{\exp(-\frac{1}{2\sigma^2}(x + \sqrt{E_b})^2)}{\exp(-\frac{1}{2\sigma^2}(x - \sqrt{E_b})^2)} = -\frac{2\sqrt{E_b}}{\sigma^2} \mathbf{c}_i \quad (3.4)$$

校验节点  $j$  到信息节点  $i$  的额外信息用似然比来表示

$$E_{j,i} = L(P_{j,i}^{ext}) \quad (3.5)$$

$$= \log \frac{1 - P_{j,i}^{ext}}{P_{j,i}^{ext}} \quad (3.6)$$

$$= 2 \tanh^{-1} \prod_{i' \in B_j, i' \neq i} \tanh(M_{j,i'}/2) \quad (3.7)$$

其中

$$M_{j,i'} \triangleq L(P_{j,i'}) = \log \frac{1 - P_{j,i'}}{P_{j,i'}} \quad (3.8)$$

信息节点  $i$  除了接收输入的 LLR 以外，还接收来自相连接的校验节点的 LLR。故信息节点  $i$  总的 LLR 为

$$L_i = L(P_i) = R_i + \sum_{j \in A_i} E_{j,i} \quad (3.9)$$

对于从信息节点发送到校验节点的信息，记为  $M_{j,i}$

$$M_{j,i} = R_i + \sum_{j' \in A_i, j' \neq j} E_{j',i} \quad (3.10)$$

故和积算法的具体步骤为

1. 给信息节点发送到校验节点的信息  $M_{j,i}$  赋值为  $R_i$ ;
  2. 计算校验节点发送到信息节点的信息  $E_{j,i}$ ;
  3. 计算信息节点的 LLR,  $L_i$ 。生成预测码字  $\hat{c}$ ，代入校验方程，若满足，则停止算法，或者达到最大迭代次数，停止算法；
  4. 计算信息节点发送到校验节点的信息  $M_{j,i}$ ，遍历次数加一。继续第二步。
- 如果该算法收敛，经过足够多次迭代后，将渐近求出码字中各位为 1 或者 0 的概率，进而实现逐符号最大后验概率译码。

## 第二节 LDPC 卷积码的滑动窗口解码

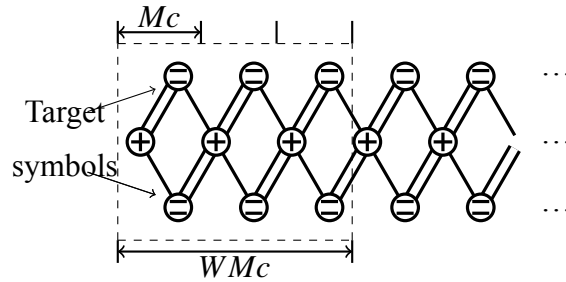


图 3.1 滑动窗口解码器的例子， $t = 0$

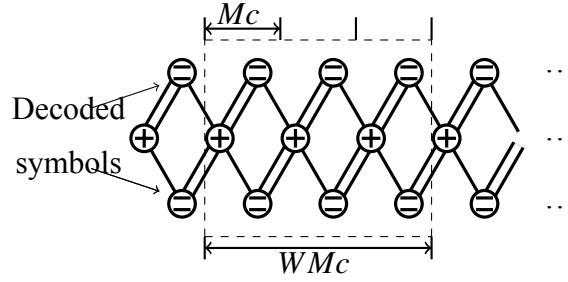

 图 3.2 滑动窗口解码器的例子,  $t = 1$ 

图3.1与图3.2为滑动窗口解码器的一个例子。其窗口大小为  $W = 3$ ，作用在  $m_s = 1$  的 (3,6) 正则  $q$  元 LDPC 卷积码上。对于窗口里的  $WMc$  个符号，解码算法不断迭代，直到某一固定迭代次数或者满足某些停止规则。然后窗口平移  $Mc$  个位置，即已解码的  $Mc$  个符号从窗口移出。滑动窗口解码的解码延时为

$$T_{CC} = WMmc \quad (3.11)$$

本文基于，LDPC 卷积码的解码延时  $T_{CC}$  等于 LDPC 分组码的解码延时  $T_{BC}$ ，的前提下，研究 LDPC 卷积码及 LDPC 分组码的性能。滑动窗口解码器的窗口内迭代算法可以使用之前描述的和积算法，也可以用和积算法的改进版本 FFT-QSPA 等。

窗口内迭代算法的停止规则如下。给定 LDPC 卷积码以及接收到的码字，令  $P_t^{(j)}(b)$  为在时刻  $t$ ，窗口中第  $j$  个符号  $v_t^{(j)}$  的  $b \in \text{GF}(q)$  的概率。经过每次和积算法的迭代，基于  $P_t^{(j)}(b)$  对  $v_t^{(j)}$  进行硬判决，记为  $\hat{v}_t^{(j)}$ ，即选择  $\hat{v}_t^{(j)} = x$  使得概率为最大值。记  $e_t^{(j)}$  为  $v_t^{(j)}$  不等于  $\hat{v}_t^{(j)}$  的概率，则

$$e_t^{(j)} = 1 - P_t^{(j)}(x = \hat{v}_t^{(j)}) \quad (3.12)$$

而对于整个窗口内的符号， $e_t^{(j)}$  的平均值为

$$\hat{P}_t = \frac{1}{Mc} \sum_{j=0}^{Mc-1} e_t^{(j)} \quad (3.13)$$

当迭代次数达到最大值  $I_{max}$ ，或者  $\hat{P}_t$  小于某个选定的误符号率时，解码窗口才平移到下一个位置。

## 第四章 模拟仿真结果

本文模拟仿真的环境为翻转概率为  $\varepsilon$  的 BSC 信道。对于  $q$  元 LDPC 分组码, 采用 QSPA 解码算法, 停止规则为最大迭代次数  $I_{max} = 100$ 。对于  $q$  元 LDPC 卷积码, 滑动窗口解码使用 QSPA 解码算法, 停止规则为最大迭代次数  $I_{max} = 100$  或者误比特率 BER 达到  $10^{-4}$ 。

### 第一节 LDPC 分组码码长与 LDPC 卷积码限制长度相等

构造  $q$  元 LDPC 分组码, 使其码长  $n_{BC} = 2Mc$ 。构造  $q$  元 LDPC 卷积码, 其限制长度  $v_s = (m_s + 1)Mc = 2Mc$ 。在  $n_{BC} = v_s = 2Mc$  的条件下进行性能上的比较。

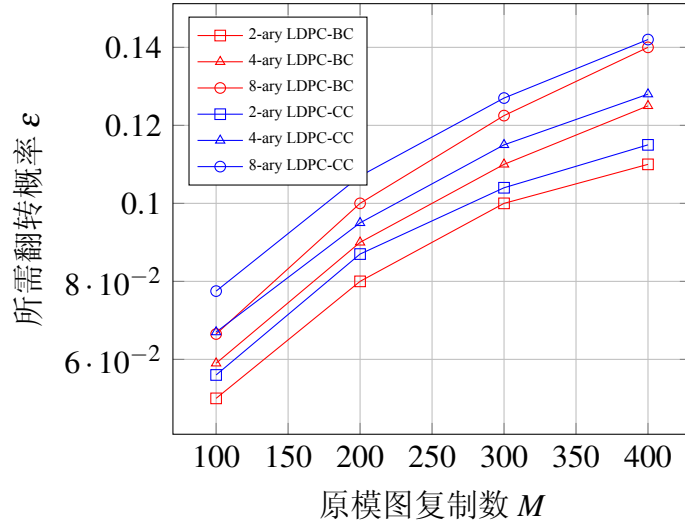


图 4.1 使用  $R = 1/2$  的 (2,4) 正则原模图构造

图4.1展示的是  $q$  元 LDPC 分组码及  $q$  元 LDPC 卷积码解码时误比特率 BER 达到  $10^{-4}$  所需翻转概率  $\varepsilon$  与原模图复制数  $M$  之间的关系。 $q$  元 LDPC 分组码及  $q$  元 LDPC 卷积码都由码率为  $R = 1/2$  的 (2,4) 正则原模图构造。 $q$  元 LDPC 卷积码的滑窗解码器的窗口大小为  $W = 12$ 。

从图4.1可以看出,  $q$  元 LDPC 分组码及  $q$  元 LDPC 卷积码的性能都随着原模图复制数  $M$  增大而提高。同时复制数  $M$  较小时, 相对于  $q$  元 LDPC 分组码,

$q$  元 LDPC 卷积码获得“卷积优势”，但是这种优势随  $M$  增大而减小。比如当  $M = 100$  时，8 元 LDPC 卷积码所需的  $\varepsilon$  比 8 元 LDPC 分组码的  $\varepsilon$  大 0.01，而  $M = 400$  时，8 元 LDPC 卷积码的  $\varepsilon$  仅比 8 元 LDPC 分组码的  $\varepsilon$  大 0.001。

这个结果与 LDPC 卷积码阈值渐近分析（即  $M$  趋向无穷）相一致，也即 LDPC 卷积码的解码阈值会随着  $M$  的增大，逐渐趋向用于构建该 LDPC 卷积码的 LDPC 分组码的解码阈值。

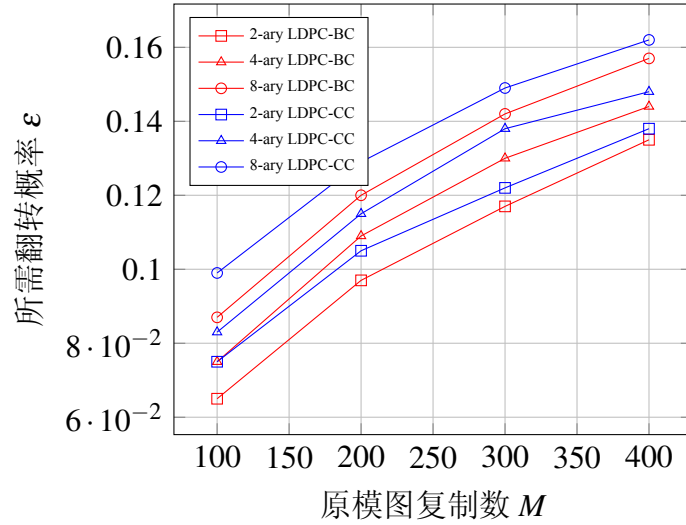


图 4.2 使用  $R = 1/2$  的 (3,6) 正则原模图构造

图4.2描述的是  $q$  元 LDPC 分组码及  $q$  元 LDPC 卷积码解码时误比特率 BER 达到  $10^{-4}$  所需翻转概率  $\varepsilon$  与原模图复制数  $M$  之间的关系。 $q$  元 LDPC 分组码及  $q$  元 LDPC 卷积码都由  $R = 1/2$  的 (3,6) 正则原模图构造。图4.2与图4.1类似，都表现了  $q$  元 LDPC 分组码及  $q$  元 LDPC 卷积码的性能随着  $M$  增大而提高的特性。同时  $q$  元 LDPC 卷积码的“卷积优势”随  $M$  增大而减小。

另外，对比图4.2与图4.1可以发现，使用 (3,6) 正则原模图比使用 (2,4) 正则原模图构造 LDPC 码的解码性能更好。

## 第二节 LDPC 分组码与 LDPC 卷积码的解码延时相等

除了解码性能以外，信道编码的解码延时对于设计高速通信系统来说十分重要。无线通信如 Wi-Fi，实时语音或视频通信，军事通信系统等对解码延时的要求比较苛刻。下面考虑  $q$  元 LDPC 分组码解码延时等于  $q$  元 LDPC 卷积码解码延时的情况。 $q$  元 LDPC 分组码的延时  $T_{BC} = M_{BC}mc$ ，及  $q$  元 LDPC 卷积码的

延时  $T_{CC} = WM_{CC}mc$ 。为使  $T_{BC} = T_{CC}$ ，只需令  $M_{BC} = WM_{CC}$ 。

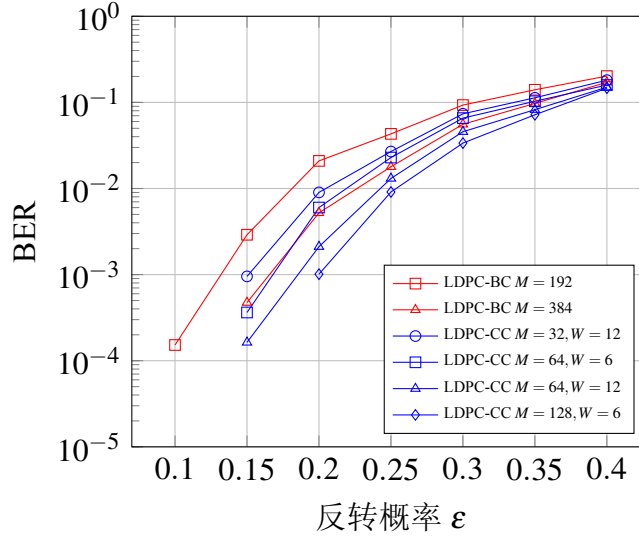


图 4.3 使用  $R = 1/2$  的 (3,6)8 元正则原模图构造

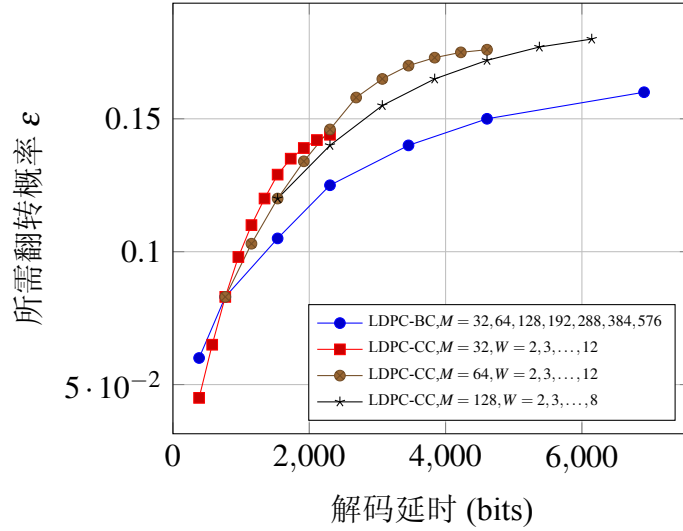


图 4.4 解码延时与所需翻转概率之间的关系

图4.3中 8 元 LDPC 分组码及 8 元 LDPC 卷积码都由码率为  $R = 1/2$  的 (3,6) 正则原模图构造。从图4.3可以看出，8 元 LDPC 卷积码的性能优于 8 元 LDPC 分组码。另外，从图4.3可以看出，用参数为  $M_{CC} = 64, W = 6$  解码器解码 LDPC 卷积码的性能优于用参数为  $M_{CC} = 32, W = 12$  解码 LDPC 卷积码的性能；用参数为  $M_{CC} = 128, W = 6$  解码器解码 LDPC 卷积码的性能优于用参数为  $M_{CC} = 64, W = 12$  解码 LDPC 卷积码的性能 (它们的解码延时相等)。也就是说，

选择相对小的窗口  $W$ ，同时选择大的复制数  $M_{CC}$  补偿解码延时，能获得更好的解码性能。

图4.4在 GF(8) 上比较要达到相同的误比特率所需的翻转概率与解码延时之间的关系。其中 8 元 LDPC 分组码及 8 元 LDPC 卷积码都由码率为  $R = 1/2$  的 (3,6) 正则原模图构造。从图4.4可以看出，当固定原模图复制数  $M_{CC}$  时，LDPC 卷积码性能随窗口大小  $W$  增加而提高，但是其解码延时  $T_{CC}$  却增加了。另外，在一定范围内大的复制数  $M_{CC}$  与小的窗口大小  $W$  的组合有更好的性能。比如，解码延时  $T_{CC} = 2304$  时， $M_{CC} = 64$  与  $W = 6$  的 LDPC 卷积码的性能比  $M_{CC} = 32$  与  $W = 12$  的 LDPC 卷积码的性能更好。然而，复制数  $M_{CC}$  越大与窗口  $W$  越小不一定有最好性能。比如，解码延时  $T_{CC} = 2304$  时， $M_{CC} = 128$  与  $W = 3$  的 LDPC 卷积码的性能反而比  $M_{CC} = 64$  与  $W = 6$  的 LDPC 卷积码的性能差。由此可见，解码延时确定时，要综合考虑复制数  $M_{CC}$  与窗口大小  $W$  的影响。

表 4.1 在不同 GF( $q$ ) 与解码延时的条件下使误比特率达到  $10^{-4}$  所需的翻转概率

解码 延时	LDPC-BC			LDPC-CC		
	GF(2)	GF(4)	GF(8)	GF(2)	GF(4)	GF(8)
2304 bits	0.14	0.15	0.15	0.12	0.16	0.18
4608 bits	0.16	0.17	0.16	0.18	0.19	2.0
6912 bits	0.17	0.18	0.17	0.19	0.21	0.21
9216 bits	0.17	0.19	0.19	0.21	0.22	0.22

表4.1展示了，(3,6) 正则  $q$  元 LDPC 分组码与 (3,6) 正则  $q$  元 LDPC 卷积码在不同的 GF( $q$ ) 以及不同的解码延时的条件下，误比特率 BER 达到  $10^{-4}$  所需翻转概率  $\epsilon$ 。从表4.1可以看出，当解码延时确定时，非二元的 LDPC 卷积码的解码性能优于  $q$  元 LDPC 分组码以及二元 LDPC 卷积码的解码性能。与  $q$  元 LDPC 分组码相反， $q$  元 LDPC 卷积码误比特率 BER 达到  $10^{-4}$  所需翻转概率  $\epsilon$  随  $q$  的增加而增大。这种现象与 [15] 中的迭代译码阈值结果相一致，即，当  $q$  增加时，(3,6) 正则  $q$  元 LDPC 卷积码的解码阈值逐趋向到信道容量，而 (3,6) 正则  $q$  元 LDPC 分组码的解码阈值开始偏离信道容量。我们还能从表中发现，对于解码延时为 2304 比特，(3,6) 正则二元 LDPC 卷积码误比特率 BER 达到  $10^{-4}$  所需翻转概率  $\epsilon$ ，要低于 (3,6) 正则二元 LDPC 分组码的翻转概率。这是由于限制长度短的二元 LDPC 卷积码存在错误扩散现象。在高误比特率或者更大解码延时的条件下，以上的错误扩散现象将会消失，从表中也可以发现，当解码延时大于

2304 比特时，二元 LDPC 卷积码解码性能比二元 LDPC 分组码更好。

### 第三节 LDPC 分组码与 LDPC 卷积码的解码复杂度比较

Pusane 等人在 [16] 中分析了，相对于二元 LDPC 分组码，使用流水线解码算法的二元 LDPC 卷积码而获得的卷积特性与计算复杂度，处理器复杂度，解码器所需内存大小以及解码延时的关系。

在这一节中，我们将在相同解码延时或者相同的解码性能的条件下，比较  $q$  元 LDPC 卷积码与  $q$  元 LDPC 分组码的解码计算复杂度。对于使用 QSPA 进行解码的  $q$  元 LDPC 码，在每次迭代周期中，一个校验节点所需的计算复杂度为  $\mathcal{O}(qm)$ ，一个变量节点所需的计算复杂度为  $\mathcal{O}(q)$ 。令  $I_{BC}$  代表要解码一个 LDPC 分组码的码字所需要的平均迭代次数，类似的，令  $I_{CC}$  代表，在某一确定时间点，在 LDPC 卷积码的窗口解码器中一个窗口内，要解码所有符号所需要的平均迭代次数。对于  $(d_v, d_c)$  正则 LDPC 分组码来说，其设计码率为  $R = \frac{d_c - d_v}{d_c}$ ，其解码每个码字所需计算复杂度为

$$\mathcal{O}\left(\frac{T_{BC}}{m}d_vq + \frac{T_{BC}}{m}(1-R)d_cqm\right)I_{BC} = \mathcal{O}\left(\left(\frac{d_v}{m} + d_v\right)qT_{BC}\right)I_{BC} \quad (4.1)$$

因此，对于解码  $(d_v, d_c)$  正则 LDPC 分组码的每一比特，所需计算复杂度为

$$\mathcal{O}\left(\left(\frac{d_v}{m} + d_v\right)q\right)I_{BC} \quad (4.2)$$

对于  $(d_v, d_c)$  正则 LDPC 卷积码，为了简单起见，我们认为在窗口解码器的窗口中的那一部分 Tanner 图是  $(d_v, d_c)$  正则的，虽然在窗口起始部位的校验节点以及结束部位的变量节点有相对低的度分布。因此，对于每个窗口，其计算复杂度大约为

$$\mathcal{O}\left(\left(\frac{d_v}{m} + d_v\right)qT_{CC}\right)I_{CC} \quad (4.3)$$

而注意到在某一特定时间，窗口解码器要解码的比特数是  $T_{CC}/W$ ，所以对于  $(d_v, d_c)$  正则 LDPC 卷积码每一个比特的解码计算复杂度是

$$\frac{\mathcal{O}\left(\left(\frac{d_v}{m} + d_v\right)qT_{CC}\right)I_{CC}}{T_{CC}/W} = \mathcal{O}\left(\left(\frac{d_v}{m} + d_v\right)q\right)WI_{CC} \quad (4.4)$$

通过比较 LDPC 卷积码与 LDPC 分组码解码每一个比特所需的计算复杂度，我们可以发现，如果  $I_{BC} = WI_{CC}$ ，在相同的  $\text{GF}(q)$  上，LDPC 卷积码与 LDPC 分



组码会有相同的解码计算复杂度。在以下的实证分析中，我们所基于的原模图是 (3,6) 正则 LDPC 码。具有其它度分布的原模图也能得出以下类似的实验结果。另外，在解码 LDPC 卷积码时，我们将窗口参数设置为  $W = 6$ 。

表 4.2 在不同  $GF(q)$  与解码延时的条件下使误比特率达到  $10^{-4}$  所需的迭代次数

解码 延时	$I_{BC}$			$I_{CC}(W = 6)$		
	GF(2)	GF(4)	GF(8)	GF(2)	GF(4)	GF(8)
4608 bits	13.8	12.3	11.1	3.3	3.2	3.0
6912 bits	15.6	14.1	12.6	3.9	3.7	3.4

在本节中，我们基于相等的解码延时的条件，比较  $q$  元 LDPC 卷积码与  $q$  元 LDPC 分组码的计算复杂度。表4.2展示了，在解码延时分别为 4608, 6912 比特时，(3,6) 正则  $q$  元 LDPC 卷积码与 (3,6) 正则  $q$  元 LDPC 分组码解码误比特率达到  $10^{-4}$  所需的平均迭代次数  $I_{CC}$  和  $I_{BC}$ 。我们可以发现，在相同的  $GF(q)$  中， $q$  元 LDPC 分组码的  $I_{BC}$  高于  $q$  元 LDPC 卷积码的  $I_{CC}$ 。这个现象的解释显而易见：对于给定的解码延时，LDPC 分组码需要解码的符号数是 LDPC 卷积码需要解码的符号数的  $W$  倍。我们还可以看到，对于 LDPC 分组码与 LDPC 卷积码，所需的迭代次数都随着  $q$  的减小而减小；但是，总的解码计算复杂度却增加了，这是因为每次迭代的复杂度增加得更多。

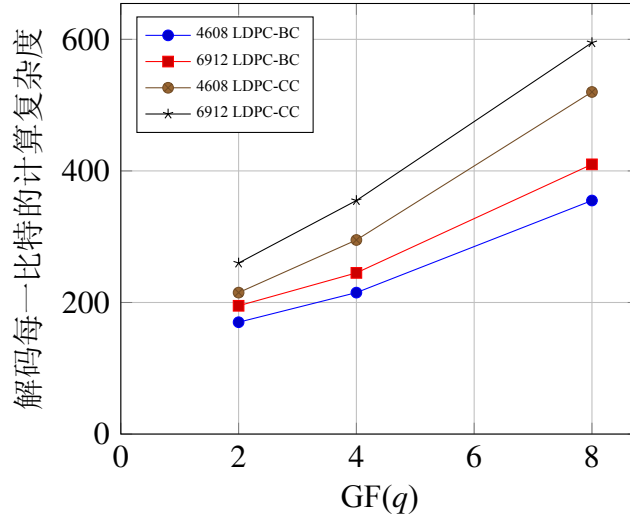


图 4.5 每解码一比特的计算复杂度与  $GF(q)$  的关系

图4.5展示了，(3,6) 正则  $q$  元 LDPC 卷积码与 (3,6) 正则  $q$  元 LDPC 分组码每解码一比特的计算复杂度与  $GF(q)$  的关系。我们可以观察到，LDPC 卷积码

与 LDPC 分组码的解码计算复杂度都随  $q$  的增大而指数增加。另外，在相同的  $GF(q)$  中，LDPC 卷积码的复杂度比 LDPC 分组码的复杂度高 35%。同时我们还可以看到，在相等解码延时的条件下，二元 LDPC 卷积码的复杂度比 4 元 LDPC 分组码的复杂度高 10%，而 4 元 LDPC 卷积码的复杂度比二元 LDPC 分组码的复杂度高 80%。但是，二元 LDPC 卷积码解码时误比特率达到  $10^{-4}$  所需翻转概率，比 4 元 LDPC 分组码所需翻转概率大 0.01，而 4 元 LDPC 卷积码所需翻转概率比二元 LDPC 分组码所需翻转概率大 0.04（见表4.1）。所以，即使 LDPC 卷积码相对于 LDPC 分组码有较高的计算复杂度，但是它性能的增加量比 LDPC 分组码更加明显。况且，很难通过增加 LDPC 分组码的解码复杂度来增加其解码性能，也就是说，简单增加解码迭代次数，LDPC 分组码也很难减小解码性能上与 LDPC 卷积码的差距。

因此，根据以上分析可以得出结论：对于给定解码延时的情况，在误码率性能和计算复杂度的平衡之间，LDPC 卷积码是更好的选择。

## 第五章 总结

在本文章中，我们主要的研究对象是基于原模图构造的有限长 LDPC 卷积码以及 LDPC 分组码。为了公平地比较了两者的性能，本文在第二章通过类似的构造方式，即基于原模图复制和连接，构造了 LDPC 分组码和 LDPC 卷积码。本文在第三章简单描述了用于解码 LDPC 分组码的和积算法，并介绍了用于解码 LDPC 卷积码的滑动窗口算法。接着，我们根据置信传播算法的软判决方法，为本文的实证分析设计了用于滑动窗口解码器的迭代停止规则。

本文的实证分析表明，在 LDPC 卷积码的限制长度与 LDPC 分组码的码长相等时，(2,4) 正则以及 (3,6) 正则  $q$  元 LDPC 卷积码相比于对应的 LDPC 分组码获得了某种“卷积优势”。在固定解码延时的条件下，我们还比较了  $q$  元 LDPC 卷积码与 LDPC 分组码的原模图复制数、解码窗口大小以及误比特率等参数之间的关系。在该条件下，实验结果表明，(3,6) 正则非二元 LDPC 卷积码的性能优于  $q$  元 LDPC 卷积码以及二元 LDPC 卷积码的性能。除此以外，在固定  $GF(q)$  以及解码延时的条件下，(3,6) 正则  $q$  元 LDPC 卷积码的解码性能随着解码窗口大小达到  $W = 6$  而达到峰值，然后解码性能随着解码窗口继续增大而渐渐下降。这种实验现象在 (2,4) 正则  $q$  元 LDPC 卷积码上也有体现。

在文章的最后，我们还分别计算并分析了  $q$  元 LDPC 卷积码与  $q$  元 LDPC 分组码的解码计算复杂度。接着，本文在相等解码延时的条件下，比较了两者的解码计算复杂度。实验结果表明，(3,6) 正则 4 元 LDPC 卷积码具有较小的解码延时，较低的解码计算复杂度，以及较好的解码性能等优点。

本文的实证分析将为以后 LDPC 卷积码进一步应用提供参考依据。

## 参考文献

- [1] R. Gallager. “Low-density parity-check codes” [J]. *IRE Transactions on Information Theory*, 1962, 8(1): 21 ~ 28.
- [2] T. J. Richardson, M. A. Shokrollahi and R. L. Urbanke. “Design of capacity-approaching irregular low-density parity-check codes” [J]. *IEEE Transactions on Information Theory*, 2001, 47(2): 619 ~ 637.
- [3] M. C. Davey and D. MacKay. “Low-density parity check codes over  $GF(q)$ ” [J]. *IEEE Communications Letters*, 1998, 2(6): 165 ~ 167.
- [4] L. Barnault and D. Declercq. “Fast decoding algorithm for LDPC over  $GF(2q)$ ” [C]. ITW’2013: *Proceedings of IEEE Information Theory Workshop*. Paris, Mar., 2003: 70 ~ 73.
- [5] H. Wymeersch, H. Steendam and M. Moeneclaey. “Log-domain decoding of LDPC codes over  $GF(q)$ ” [C]. ICC’2004: *Processings of IEEE International Conference on Communications*. San Diego, Jan., 2004: 772 ~ 776.
- [6] A. J. Felstrom and K. S. Zigangirov. “Time-varying periodic convolutional codes with low-density parity-check matrix” [J]. *IEEE Transactions on Information Theory*, 1999, 45(6): 2181 ~ 2191.
- [7] S. Kudekar, T. J. Richardson and R. L. Urbanke. “Threshold saturation via spatial coupling: why convolutional LDPC ensembles perform so well over the BEC” [J]. *IEEE Transactions on Information Theory*, 2011, 57(2): 803 ~ 834.
- [8] D. J. Costello et al. “A comparison of ARA- and protograph-based LDPC block and convolutional codes” [C]. ITA’2007: *Processings of Information Theory and Applications Workshop*. Paris, June, 2007: 111 ~ 119.
- [9] A. E. Pusane et al. “Implementation aspects of LDPC convolutional codes” [J]. *IEEE Transactions on Communications*, 2008, 56(7): 1060 ~ 1069.
- [10] A. R. Iyengar et al. “Windowed decoding of protograph-Based LDPC convolutional codes over erasure channels” [J]. *IEEE Transactions on Information Theory*, 2012, 58(4): 2303 ~ 2320.
- [11] N. ul Hassan, M. Lentmaier and G. P. Fettweis. “Comparison of LDPC block and LDPC convolutional codes based on their decoding latency” [C]. ISTC’2012: *Processings of 7th International Symposium on Turbo Codes and Iterative Information*. Gothenburg, Aug., 2012: 225 ~ 229.
- [12] T. Hehn and J. B. Huber. “LDPC codes and convolutional codes with equal structural delay: a comparison” [J]. *IEEE Transactions on Communications*, 2009, 57(6): 1683 ~ 1692.
- [13] J. Thorpe. “Low-Density Parity-Check (LDPC) codes constructed from protographs” [J]. *Interplanetary Network Progress Report*, 2003, 154: 1 ~ 7.

- [14] X.-Y. Hu, E. Eleftheriou and D. M. Arnold. “Regular and irregular progressive edge-growth tanner graphs” [J]. *IEEE Transactions on Information Theory*, 2005, 51(1): 386 ~ 398.
- [15] L. Wei et al. “Threshold analysis of non-binary spatially-coupled LDPC codes with windowed decoding”[C].ISIT’2014: *IEEE International Symposium on Information Theory*. Honolulu, June, 2014: 881 ~ 885.
- [16] A. E. Pusane et al. “Deriving good LDPC convolutional codes from LDPC block codes” [J]. *IEEE Transactions on Information Theory*, 2011, 57(2): 835 ~ 857.