

南 开 大 学

本 科 生 毕 业 论 文（设 计） (双学位专用)

GF(q)上 LDPC 分组码和卷积码的比较
Comparison of LDPC Block and Convolutional Codes over GF(q)

学 号： 1210403

姓 名： 孙卓豪

年 级： 2012 级

专 业： 电子信息科学与技术

系 别： 电子信息科学与技术

学 院： 电子信息与光学工程

双修专业： 数学与应用数学

双修院系： 数学科学学院

指导教师： 陈鲁生

完成日期： 二〇一六年五月

关于南开大学本科毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在指导教师指导下，进行研究工作所取得的成果。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、已公开发表或没有公开发表的作品内容。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

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

摘 要

本文介绍了 $GF(q)$ 上 LDPC 分组码，并从 LDPC 分组码构建结构相似的卷积码。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。基于以上条件比较多元 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与 2 元 LDPC 卷积码，以及 q 元 LDPC 卷积码的解码性能好与 q 元 LDPC 分组码。

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.

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
第二章 基于原模图构造的 LDPC 码	3
第一节 LDPC 码的原模图及 LDPC 码的叠加方式	3
第二节 LDPC 分组码的构造	4
第三节 LDPC 卷积码的构造	5
第三章 和积算法与滑动窗口解码的简介	8
第一节 LDPC 分组码的和积算法	8
第二节 LDPC 卷积码的滑动窗口解码	9
第四章 模拟仿真结果	11
第一节 LDPC 分组码码长与 LDPC 卷积码限制长度相等	11
第二节 LDPC 分组码与 LDPC 卷积码的解码延时相等	12
第五章 总结	15
参考文献	16

第一章 背景及介绍

Gallager 在 1962 年提出低密度校验分组码 [1]，同时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 $GF(q)$ 的 LDPC 分组码 [3]，同时把 Gallager 的 BP 解码算法推广到 q 元域，称作 q 元和积算法，并得到很好的性能。1999 年，Davey 将快速傅立叶变换引入和积算法中，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]。

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 卷积码的特点是它的 BP 解码阈值能达到它所基于的 LDPC 码的最大后验概率 (MAP) 解码阈值 [7]，简称阈值饱和现象。Pusane 等人在 [8] 专门为 LDPC 卷积码设计了并行的流水线解码算法，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，将占用大量内存，同时有很高的译码延时。为了解决 LDPC 卷积码顺序解码时以上缺点，Papaleo 等人考虑了 BP 算法窗口解码版本 [9]，并研究了算法中解码延时和解码性能之间的取舍。

在构建 LDPC 卷积码方面，Felström 和 Zigangirov 做出了开创性工作 [6]。随后基于原模图 [10] 构造 LDPC 卷积码的方法被提出。结合原模图与 progressive edge growth 算法 [11]，LDPC 卷积码能很简单地构造出来。

由于 LDPC 卷积码与 LDPC 分组码具有相似的结构，Costello 在 [12] 与 [13] 中比较了二元 LDPC 分组码与二元 LDPC 卷积码的性能。Hassan 等人基于解码延时 [14] 和 Hehn 等人基于结构延时 [15] 研究 LDPC 分组码与 LDPC 卷积码的特点。本文基于以上的研究进行 q 元域的推广，同时比较其他相关参数对 LDPC 卷积码与 LDPC 分组码解码性能的影响。

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。第四章通过调整几类参数，比较 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

每一个 LDPC 码都对应着一个 Tanner 图。在 LDPC 码的叠加过程中，该 Tanner 图称为原模图，或者基本图。如图 2.1，该图右侧为对应 LDPC 码的校验矩阵，其中圈中等号代表变量节点，圈中加号代表校验节点。

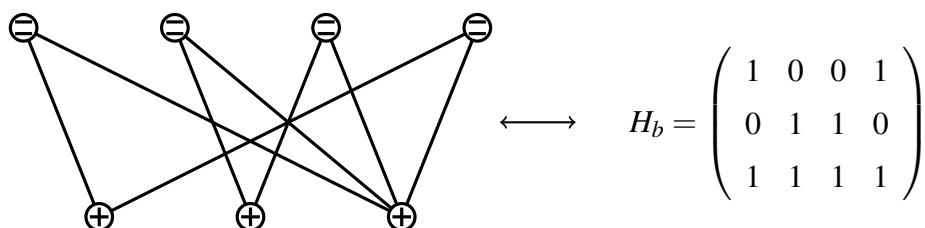


图 2.1 基本图

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边相连接，则可以将基本图折叠，如图 2.2 为 (3,6)-正则 LDPC 码的原模图以及对应的基本矩阵描述。

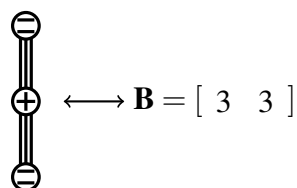
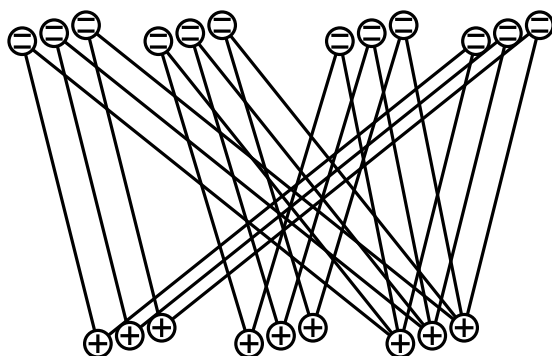


图 2.2 LDPC 码的原模图

另外，可以通过叠加基本图构造有结构的 LDPC 码。比如，先复制 m 次基本图，得到簇状的 Tanner 图，如 2.3。


 图 2.3 复制 m 次基本图

然后将基本图对应的 H_b 中为 1 的元素换为来自置换矩阵集合的元素。即

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

其中 p^i 对应循环置换矩阵，比如

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

图2.4即为以上方法构造出来的具有一定结构的 LDPC 码。

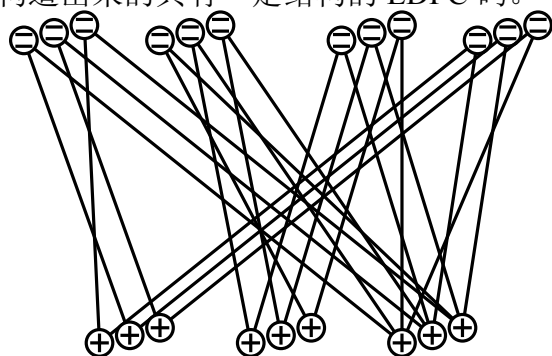


图 2.4 具有一定结构的 LDPC 码

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。 $\text{GF}(q)$

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。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}_{SC}

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

其中 m_s 在示意图中为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}_{SC} 中的非零元替换为随机选择的 $M \times M$ 置换矩阵，将 \mathbf{B}_{SC} 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}_{SC} 。最后将 \mathbf{H}_{SC} 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}_{SC} ，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 $\mathbf{B}_0 = \begin{bmatrix} 2 & 1 \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} 1 & 2 \end{bmatrix}$ 。

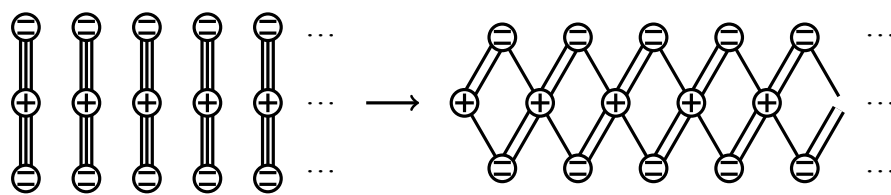


图 2.5 构造 LDPC 卷积码

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 $c_{i'} = 1$ 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 $c_i = 0$ 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 $\mathbf{c}_i = 0, 1$ ，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。或者达到最大遍历值停止算法。
 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

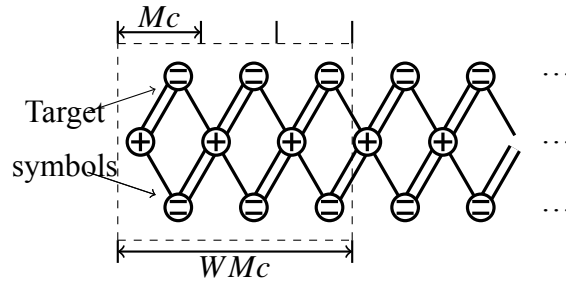


图 3.1 滑动窗口解码器的例子， $t = 0$

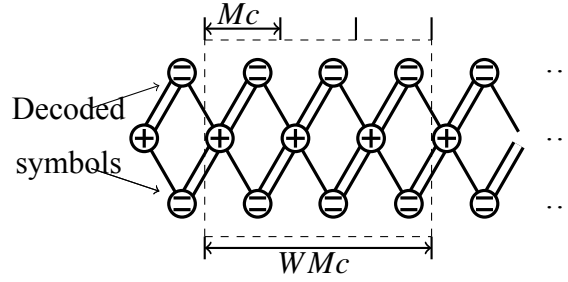

 图 3.2 滑动窗口解码器的例子, $t = 1$

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 ε 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^{-5} 。

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。

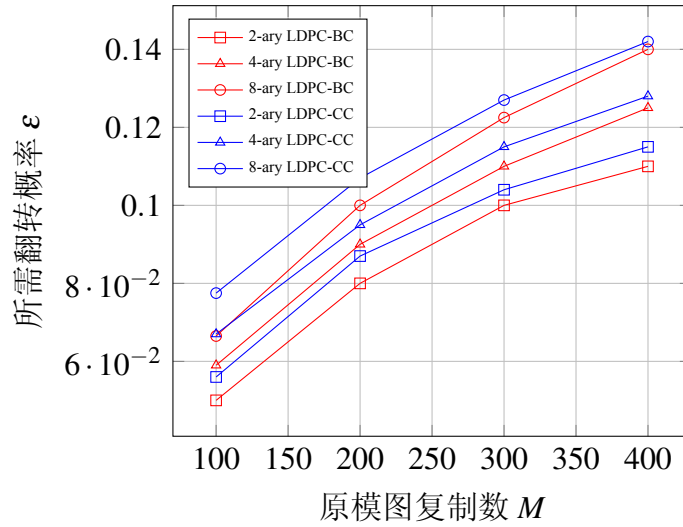


图 4.1 使用 $R = 1/2$ 的 (2,4) 正则原模图构造

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 ε 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 ε 比 8 元 LDPC 分组码的 ε 大 0.01，而 $M = 400$ 时，8 元 LDPC 卷积码的 ε 仅比 8 元 LDPC 分组码的 ε 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。

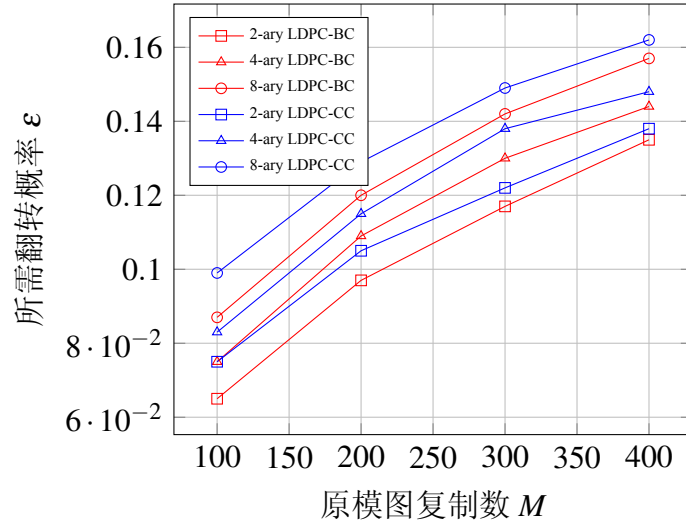


图 4.2 使用 $R = 1/2$ 的 (3,6) 正则原模图构造

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 ε 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 $M_{BC} = WM_{CC}$ 。

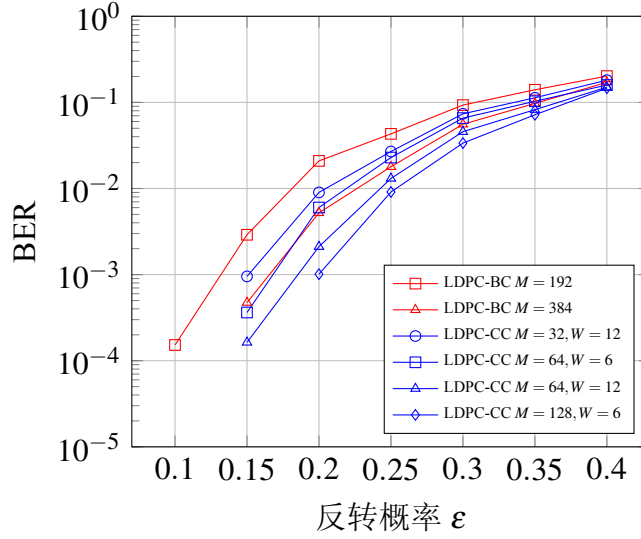


图 4.3 使用 $R = 1/2$ 的 (3,6)8 元正则原模图构造

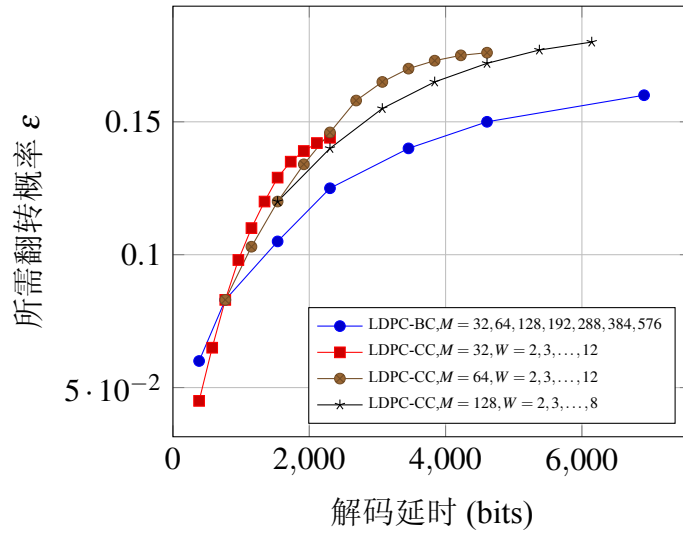


图 4.4 解码延时与所需翻转概率之间的关系

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，由 $M_{CC} = 64, W = 6$ 生成的 LDPC 卷积码的性能优于由 $M_{CC} = 32, W = 12$ 生成的 LDPC 卷积码，由 $M_{CC} = 128, W = 6$ 生成的 LDPC 卷积码的性能优于由 $M_{CC} = 64, W = 12$ 生成的 LDPC 卷积码 (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。

第五章 总结

本文在 $\text{GF}(q)$ 上，基于 LDPC 分组码以及基于原模图构造的有限长 LDPC 卷积码展开讨论，并比较了两者的性能。

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，并保证两者的解码延时相等。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。

通过实现以上算法进行解码实验，模拟结果表明 LDPC 卷积码所隐含的卷积特性使其性能超过 LDPC 分组码。另外，本文还比较了，原模图复制数、解码窗口大小以及误比特率等参数之间的联系。从中，我们可以得到多元 LDPC 卷积码性能比二元 LDPC 卷积码、多元 LDPC 分组码的性能更好。在解码延时确定的条件下，原模图复制数、解码窗口大小等参数的组合导致不同的误比特率，故应根据信道及校验矩阵等具体情况确定最优组合。本文将为以后的应用提供参考数据。

参考文献

- [1] R. Gallager. *Low-density parity-check codes* [M]. M.I.T. Press, **1963**: 143 ~ 148.
- [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): 70 ~ 71.
- [4] M. C. Davey. “Error-Correction using Low Density Parity Check Codes” [J]. *Ph D Dissertation University of Cambridge*, **1999**.
- [5] H. Wymeersch, H. Steendam and M. Moeneclaey. “Log-domain decoding of LDPC codes over $GF(q)$ ” [C]. In: *Communications, 2004 IEEE International Conference on*, **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. Richardson and R. Urbanke. “Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC” [C]. In: *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, **2010**: 803 ~ 834.
- [8] A. E. Pusane et al. “Implementation aspects of LDPC convolutional codes” [J]. *Communications IEEE Transactions on*, **2008**, 56(7): 1060 ~ 1069.
- [9] M. Papaleo et al. “Windowed erasure decoding of LDPC Convolutional Codes” [C]. In: *Information Theory (ITW 2010, Cairo), 2010 IEEE Information Theory Workshop on*, **2010**: 1 ~ 5.
- [10] J. Thorpe. “Low-Density Parity-Check (LDPC) Codes Constructed from Protographs” [J]. *Ipn Progress Report*, **2003**, 154.
- [11] X. Y. Hu, E. Eleftheriou and D. M. Arnold. “Regular and irregular progressive edge-growth tanner graphs”. *IEEE Transactions on Information Theory*, **2005**, 51(1): 386 ~ 398.
- [12] D. J. Costello et al. “A comparison between LDPC block and convolutional codes” [J]. *Proc inform theory and App workshop*, **2006**.
- [13] D. J. Costello et al. “A comparison of ARA and protograph-based LDPC block and convolutional codes” [J]. *Information Theory and Applications Workshop*, **2007**: 2007.
- [14] N. Ul Hassan, M. Lentmaier and G. P. Fettweis. “Comparison of LDPC block and LDPC convolutional codes based on their decoding latency” [C]. In: *Turbo Codes and Iterative Information Processing (ISTC), 2012 7th International Symposium on*, **2012**: 225 ~ 229.

- [15] 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.