

代 号 10701

学 号 0808120250

分类号 TN929.5

密 级 公开

西安电子科技大学

硕士学位论文



题 (中、英文) 目 LT 码的非等差错保护方法研究

LT Codes with Unequal Error Protection Encoding

作者姓名 种雕雕 指导教师姓名、职称 陈彦辉 教授

学科门类 工学 学科、专业 通信与信息系统

提交论文日期 二〇一一年一月

西安电子科技大学

学位论文独创性（或创新性）声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切的法律责任。

本人签名：钟雕雕

日期 2011.3.8

西安电子科技大学

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律署各单位为西安电子科技大学。

本人签名：钟雕雕

日期 2011.3.8

导师签名：

日期 2011.3.8

摘要

随着互联网技术的发展和普及,网络中大规模数据分发和广播业务越来越多,对高效可靠的数字传输和存储系统的需求也日益增长。如何在有限的带宽资源上实现信息的高速可靠传输,是通信技术研究工作的重要课题。类似音视频数据的网络业务不但数据量大而且不同的信息可能具有不同的重要性,并且网络中数据的丢失率也是随时变化的,所以使用基于可变速率纠错码的非等差错保护(UEP)无疑是实现高效可靠传输的一个较好的选择。

喷泉码是一种在删除信道下性能优越的纠错码,其码率可以在传输过程中根据信道的丢包情况而变化。LT 码是喷泉码的第一次实现,编译码简单,同时 LT 码也是 Raptor 码的基础。本文主要研究了基于 LT 码的非等差错保护方法。

本文首先介绍了 LT 码的编译码原理,对 LT 码的误码性能进行仿真,并以重复信息块编码方法为例简单介绍了与或树分析法。然后,本文从误码率和数据恢复率两个角度考虑,提出了两种 LT 码的非等差错保护方法,即基于扩展窗的逐个选择法和基于度数逐个选择法。其中,基于扩展窗的逐个选择法只要给定合适的参数,就可以实现非等差错保护,其误码性能甚至可以超过 EWF 编码方法。基于度数的逐个选择法能够在保证良好误码率的条件下,经过较少的迭代次数就可以恢复出原始信息符号,即在相同的迭代次数下,该方法的数据恢复率比其他的方法高。

关键词: LT 码 非等差错保护 与或树分析法

Abstract

With the development and popularization of the Internet technology, there are more and more mass-data distribution and broadcast services in the network. The demands for efficient and reliable digital transmission and memory system are growing. Achieving high-speed and reliable transmission with limited bandwidth becomes an important research topic. Network service like voice and video data has large quantity and different importance, moreover, data loss rate in the network is constantly changing, so the use of unequal error protection (UEP) based on rateless erasure codes becomes a better choice to achieve high-speed and reliable transmission.

Fountain code is a kind of erasure code with high performance in erasure channels, and the rate may be changed according to the package loss during the transmission. LT codes are the first realization of fountain codes with simple encoding and decoding, and are the base of raptor codes. This paper focuses on the unequal error protection method based on LT codes.

Firstly, this paper introduces the encoding and decoding principles of LT codes and the bit error rate (BER) performance of LT codes is simulated. Meanwhile, the And-Or tree analysis method is illustrated briefly taking the block duplication encoding method as an example. Then, from the points of BER and the data recovery rate, this paper proposed two new UEP methods based on LT codes, separately called one by one selection method based on extended window and one by one selection method based on degree. One by one selection method based on extending window, focused on BER performance, can realize unequal error protection with appropriate parameter, and the BER performance is even better than EWF (Extending Window Fountain codes) method. While one by one selection method based on degree can recover the original information with less iterations, ensuring good BER performance. It is to say, the second method achieves a higher data recovery rate with the same iterations than other methods.

Keyword: LT codes UEP And-Or tree

目录

第一章 绪论.....	1
1.1 论文研究背景.....	1
1.2 LT码的研究现状.....	2
1.3 论文的主要工作及章节安排.....	3
第二章 LT码的相关知识介绍.....	5
2.1 删除信道.....	5
2.2 喷泉码.....	6
2.3 LT码的编译码原理.....	8
2.3.1 LT码的编码.....	8
2.3.2 LT码的译码.....	10
2.3.3 度分布.....	10
2.4 LT码性能仿真.....	13
2.4.1 LT码的编译码实现.....	13
2.4.2 LT码在BEC信道上的性能.....	14
2.5 LT码的非等差错保护.....	16
2.5.1 与或树分析法.....	17
2.5.2 重复信息块编码方法.....	18
2.6 本章小结.....	21
第三章 基于扩展窗的逐个选择法.....	23
3.1 EWF编码方法介绍.....	23
3.2 基于扩展窗的逐个选择法.....	27
3.3 基于扩展窗的逐个选择法的仿真对比.....	29
3.4 本章小结.....	31
第四章 基于度数的逐个选择法.....	33
4.1 UEP-LT编码方法介绍.....	33
4.2 基于度数的逐个选择法.....	36
4.2.1 划分度值法.....	36
4.2.2 基于度数的逐个选择法的编码方法.....	38
4.3 基于度数的逐个选择法的仿真对比.....	40
4.4 本章小结.....	42
第五章 结束语.....	45
致谢.....	47
参考文献.....	49

第一章 绪论

1.1 论文研究背景

随着互联网技术的发展与普及,端到端的抗误码和抗分组丢失技术的研究受到越来越多的关注。这种网络的信道模型可用Elias于1955年提出的二进制删除信道^[1](BEC)来描述。互联网上传输大容量的数据,如大型软件、音视频文件等,需要进行多播或者广播传输,这些传输必须是完全可靠的。实际中,互联网中传输数据包时,一般认为接收端或者正确接收到数据包,或者完全丢失。当用户请求传输的数据包丢失时,可采用请求重传(ARQ)技术,来保证数据的可靠性。实际中,通常使用TCP/IP协议,利用反馈确认信息的方式进行数据重传。这一方法有效解决了可靠传输的问题,但是在组播传输中或者信道受到严重损害的情况下,反馈重传方式容易使得网络过载和阻塞,数据传输效率低,传输时延大,还可能形成“反馈风暴”,被公认为性能不佳。

另外,大容量的数据存储设备,例如硬盘、光盘等的数据读取,类似于海量数据传输。当部分记录由于存储介质保护不当(例如硬盘磁道损坏,光盘划伤等)而丢失时,需要用相应的信道编码技术来保证数据能被恢复。

类似上述问题的解决,需要一种能够利用随机得到的信息恢复出原始信息的编码方法。这些随机数据的位置一般是已知的。通信系统中使用的前向纠错(Forward Error Correcting, FEC)技术就是利用编码的方法,把要传输的 k 比特原始数据编码为 $n(n > k)$ 比特的数据发送出去,接收端收到编码信息后,利用适当的译码方法恢复出 k 比特的原始数据,相应的码称为前向纠错码,或复损码(Loss-Resilient Code)。一般的纠错码,例如汉明码、BCH码、RS码、卷积码、Turbo码、LDPC码等,主要用在数据位置未知的信道中,而对于数据位置已知的删除信道,要用较低编译码复杂度的纠删码。纠删码主要有三类:RS类纠删码、级联低密度纠删码和数字喷泉码。

RS码是一类有很强纠错能力的多进制BCH码,首先由Reed和Solomon于1960年提出^[2],它不仅能纠正突发错误,还可纠正随机错误。RS类纠删码根据生成矩阵的不同,可以分为范德蒙码^[3]和柯西码^[4]。

为了构造一种具有低运算复杂度,并且能以接近信道容量的速率进行传输的纠错码,Spielma用基于扩展图的LDPC码,设计了具有线性编码时间复杂度和良好纠错性能的码。Luby在此基础上提出了具有这样特性的实用码——Tornado码^[5]。与RS纠删码相比,Tornado码编码和译码都只使用简单的二进制异或运算,而RS

纠删码要在有限域上进行复杂的矩阵运算,因此其编译码速度比RS纠删码快。但是,这两种码都是固定码率的纠删码,在许多要求低码率的实际应用中并不适用。而且,低密度纠删码是非确定性码,可能出现丢失很少一部分数据就不能恢复出原始数据的情况。数字喷泉码的出现为这类问题的解决提供了新方向。

数字喷泉码是M.Luby等人于1998年提出的新概念^[6],包括随机线性喷泉码、LT码^[7]和Raptor码^[8]。

所谓数字喷泉码,是一种在删除信道下性能优越的稀疏矩阵码,码率不受限,它具有如下特征:在发送端可以由一定数量的原始数据包生成任意数量的编码数据包,而接收端只要正确接收到一定数量的编码数据包,即可通过译码以较高概率成功恢复全部原始数据包,并且不用考虑接收到的是哪些数据包以及这些数据包的接收顺序。这种编码方法之所以称为喷泉码,是因为数字喷泉码的编码整个过程就像源源不断产生小水滴的喷泉,我们只要接收到足够的数量的小水滴,就可以达到饮用的目的。与传统的信道编码技术相比,数字喷泉码具有不固定码率、低编译码复杂度、适合大数据量广播应用的特点。

2002年M.Luby提出了数字喷泉码的第一次真正意义上的实现——LT码。LT码的编码和解码复杂度都较低,对信道的适应性强,在任意信道下都具有良好的误比特性能,因而逐渐受到人们的关注。

Raptor码在编码前,先对数据进行预处理,然后再用LT编码算法进行编码。Raptor码解码过程中,利用LT码的解码技术只需恢复固定比例的中间校验单元,再利用传统纠错码的解码性质,就可以恢复所有的输入信息符号,放宽了LT码解码的要求,减小了编译码的复杂度。基于上述优良特性,Raptor码已被ITU采纳为3GPP中多媒体广播多播服务和数字视频广播的前向纠错标准。

1.2 LT 码的研究现状

近年来,人们对LT码的研究更加深入,主要研究方向有:有噪信道下的LT码译码算法及性能研究^{[9][10][11]},分布式的LT码^{[12][13]}的研究,LT码的次优度分布研究^[14],用于改善LT码性能的构造法研究^{[15][16]},LT码的非等差错保护(UEP)方法研究,以及基于UEP的音视频文件传输^[17]等等。上述诸多研究方向中,基于LT码的非等差错保护方法研究,正是本文的研究内容。

传统的数据传输中,所有的数据都是等差错保护(EEP)的,但在有些情况下对数据中一部分数据的可靠性要求很高,其他数据的可靠性可以相对差一些,这时需要用到非等差错保护策略。例如多媒体数据传输中,传输一幅图像,人物往往比背景更重要,需要更多的保护。文献[18]、[19]、[20]等在LT码的非等差错

保护策略方面均有相关的研究，这些将在本文后面章节加以详细介绍。文献[21]给定权重矩阵，按照该矩阵对信息进行LT码的UEP编码，实现了对信息的非等差错保护。但是该编码方法的优劣取决于权重矩阵，而该矩阵中参数的选择有一定的随意性，故本文不对这种方法进行过多的介绍。

1.3 论文的主要工作及章节安排

本文主要研究 LT 码的非等差错保护策略，在现有的 LT 码的 UEP 方法的基础上，提出两种新的 UEP 编码方法，并通过理论分析、仿真分析验证这两种方法的可行性及优越性。这两种基于 LT 码的非等差错保护的策略为：基于扩展窗的逐个选择法和基于度数的逐个选择法。一般来说，LT 码的 UEP 方法都是通过增加高等级数据的选取概率实现的。本文提出的第一种 LT 码的 UEP 方法与现有文献中关于 LT 码的 UEP 方法均是从这方面考虑的。而本文提出的第二种方法着重考虑度数与选取概率之间的关系，目的不仅要达到非等差错保护的效果，还要使得高优先级的数据较低优先级更早恢复出来。

本文的内容安排如下：

第二章：主要介绍 LT 码在 BEC 信道下的编译码方法，通过仿真研究系统地分析和验证了度分布及信息符号长度对 LT 码性能的影响，最后介绍可用于 LT 码的 UEP 编码方法的与或树分析方法。

第三章：提出第一种新的 LT 码的 UEP 编码方法，基于扩展窗的逐个选择法。首先介绍 EWF 编码方法和基于扩展窗的逐个选择法的编码过程，并分别对其进行理论分析和仿真，最后将这两种编码方法进行仿真对比，说明本章提出方法的误码性能。

第四章：介绍 UEP-LT 编码方法，并通过仿真分析逐步提出另一种基于 LT 码的非等差错保护方法，即基于度数的逐个选择法。然后将本章提出的编码方法分别与 UEP-LT 编码方法和 EWF 编码方法进行对比，说明该方法在数据恢复方面的优势。

第五章：对全文内容进行总结，指出本文存在的欠缺并给出下一步的研究方向。

第二章 LT 码的相关知识介绍

2002 年 M.Luby 提出了一种稀疏的随机线性喷泉码——LT(Luby Transform) 码, LT 码成为喷泉码的第一次具体实现。LT 码最初是为删除信道设计的, 由于其特殊的码率无关性, 在删除信道上获得了逼近香农限的优越性能。LT 码与传统分组码不同的是其编码符号的个数不是预先确定的, 而是随需要而确定, 即编码器不断生成编码符号, 通过删除信道发送出去, 直到接收端收到足够的编码符号为止。因此 LT 码对信道有很强的适应性。后来, LT 码的应用被推广到二进制对称信道 (BSC)、加性高斯白噪声信道 (AWGN) 等信道^[9]中。本论文主要在删除信道下对 LT 码进行相关研究。

本章主要介绍 LT 码的编译码原理及 LT 码的非等差错保护的分析方法等。

2.1 删除信道

删除信道是一种重要的信道模型^[1]。互联网是第一个实际的删除信道模型, 在此之前删除信道只在理论上存在。文件在互联网中的传输是基于数据包通信的, 通常每个数据包或者无差错地被接收端接收, 或者根本没有被接收端收到。图 2.1 是一个简单的二进制删除信道(BEC)模型。

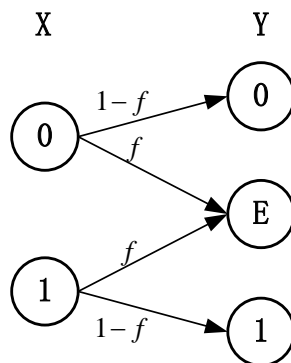


图 2.1 BEC 信道模型

图中, 有限离散信道的输入、输出概率空间分别为 $X(0,1)$, $Y(0,E,1)$, 信道的删除概率为 f 。当信宿不能决定是“0”还是“1”时, 输出一个符号 E 。“删除”是指, 遇到“ E ”就删除, 既不作“1”也不作“0”处理。

在这种信道模型下传输数据, 当数据包被丢失时, 接收端向发送端发送消息, 请求发送端重新发送该数据包。发送端收到请求重发消息, 就会立即重新发送该数据包。这种传输方式看似非常有效, 问题是这里我们没有考虑信道的删除概率。当删除概率很大时, 请求重传的信息将会很大, 发端不断重传数据包, 信道的负

载无疑会非常大。此时，使用反馈重传策略是不明智的。尤其是在具有删除特性的广播或者多播信道中，一个发送者多个接收者，当接收端都向发送端发送重传请求时，数据传输的效率更加不敢想象。所以，我们要使用一种不需要反馈或者几乎不需要反馈的编码方法，实现在删除信道上的大规模数据传输。

2.2 喷泉码

喷泉码是 John Byers 及 Michael Luby 等人于 1998 年首次提出，目的是解决大规模数据传输。喷泉码的编码过程类似于源源不断产生水滴的喷泉，每滴水就像一个编码符号。我们只要用水杯接收足够的水滴就可以饮用，同样的，我们只要接收足够的编码符号，就可以将原始信息译出来。这里的“符号”，是数据在删除信道中传输的基本单元。

喷泉码是一种无率码，可以生成无穷多的编码符号。而喷泉码的编码符号数目是在传输过程中确定的，接收端仅需要用数目稍多于原始信息符号的任意编码符号，就可以完全恢复出原始信息。这样，无论删除信道的删除概率 f 是多少，都可以根据需要不断生成编码符号，直到接收端收到足够的符号进行译码。由于 LT 码能以接近最少的编码符号进行译码，故它在任何删除信道下都是性能接近最优的编码。

最简单的喷泉码是随机线性喷泉码。假设输入符号序列为 s_1 、 s_2 、... s_K ，编码器产生 K 个随机位 $\{G_{kn}\}$ ，其中 n 是编码器的时钟周期，编码符号 t_n 由下计算：

$$t_n = \sum_{k=1}^K s_k G_{kn} \quad \text{式(2-1)}$$

编码符号在删除信道中传输，一部分符号丢失，如图 2.2 中灰色部分所示。与接收到的编码符号相对应的矩阵中的列，重新组合生成矩阵 G_{kN} 。

如图 2.2 所示，若接收到 N 个编码符号， K 和 N 满足怎样的条件才能准确恢复出输入的信息符号呢？

如果 $N < K$ ，接收端没有足够的信息，不能恢复出输入的信息符号。

如果 $N = K$ ，只要生成矩阵 G_{kN} 是可逆的，接收端可以通过高斯消去法计算出 G_{kN}^{-1} ，从而由下式计算出输入的信息符号：

$$s_k = \sum_{n=1}^N t_n G_{nk}^{-1} \quad \text{式(2-2)}$$

但是，对于任意大于 10 的 k ，随机二进制矩阵 G_{kk} 可逆的概率最大仅为 0.289^[22]。

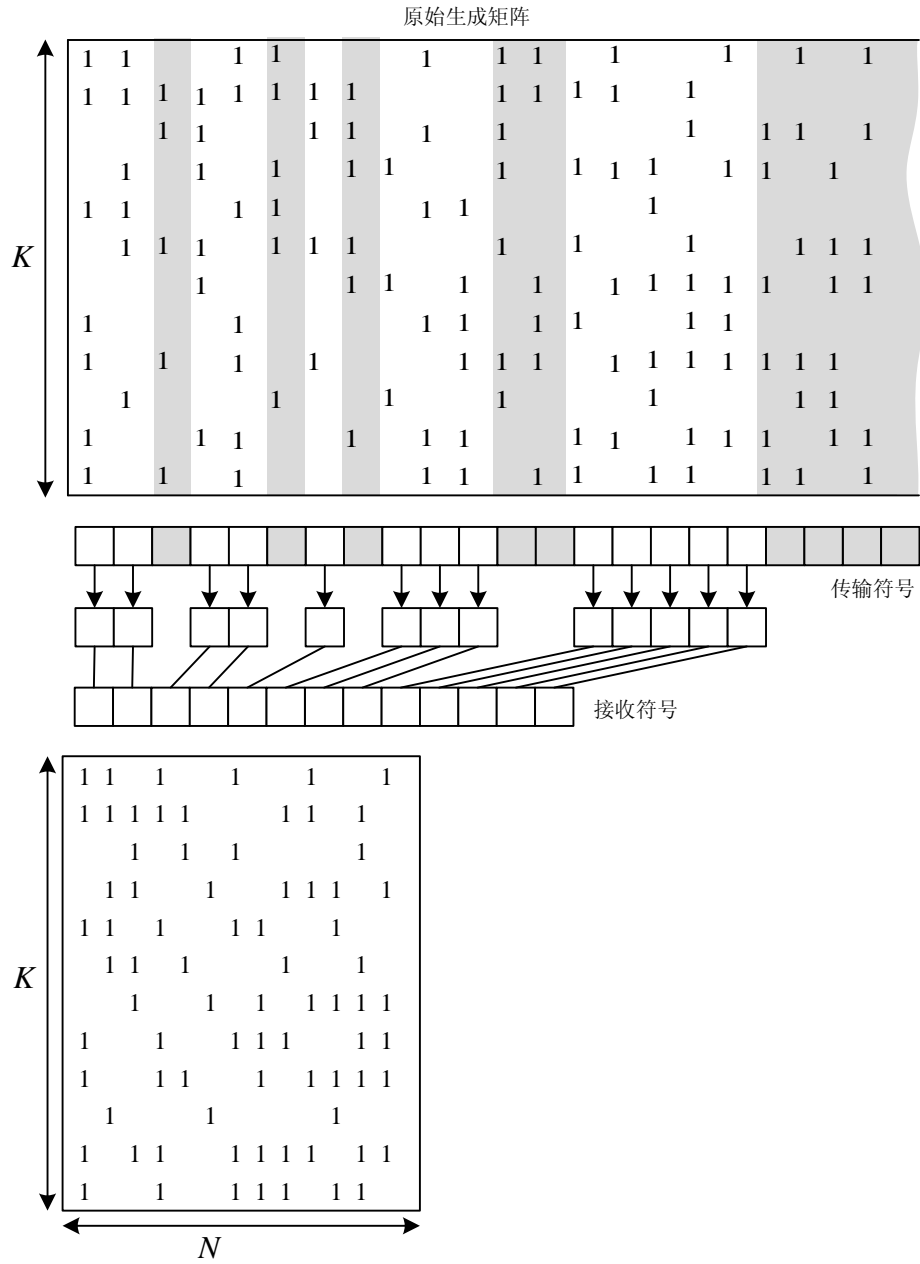


图 2.2 随机线性码的生成矩阵

如果 $N > K$ ，令 $N = K + E$ ， E 为额外接收到的编码符号。设随机二进制矩阵 G_{kN} 中包含可逆矩阵 G_{kk} 的概率为 $1 - \delta$ ， δ 为接收 N 个符号，接收端不能成功译码的概率。对于任意的 k ，译码失败的概率满足式(2-3)^[22]

$$\delta(E) \leq 2^{-E} \quad \text{式(2-3)}$$

随机线性码在多接收 E 个符号时，成功译码的概率至少为 $1 - \delta$ ， $\delta = 2^{-E}$ 。因此，随着 K 的增加，随机线性喷泉码的性能会无限接近香农限，而编译码的代价也会随之增加。故随机线性喷泉码只能用于 K 较小的情况，通常 $K < 1000$ 。

LT 码是喷泉码很重要的一类码，我们将在下一节详细介绍 LT 码。

Raptor 码采用由内外码级联的结构，内码是一个弱化的 LT 码，外码是传统的纠错编码，一般为 LDPC 码（低密度奇偶校验码）或者 RS 码。我们将外码的生成过程称为 Raptor 码的预编码。如图 2.3 所示，预编码将信息符号转换成中间编码符号，再将中间编码符号作为 LT 码的输入信息进行编码。这样，在 Raptor 码的译码过程中，LT 码只需恢复固定比例的中间编码符号，再利用外码的译码就可以恢复出所有的输入符号。

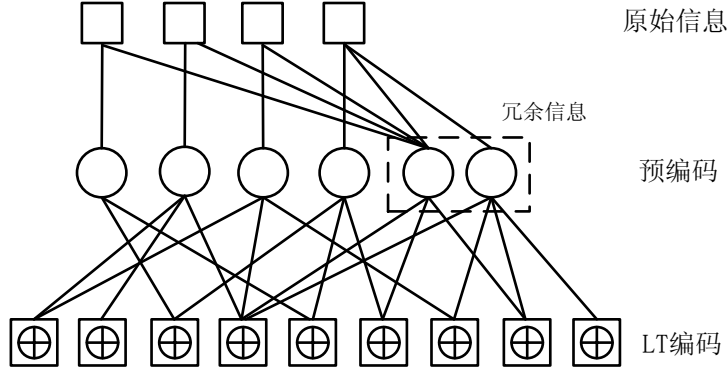


图 2.3 Raptor 码的编码示意图

由于 Raptor 码采用弱化的 LT 码作为内码，每个编码符号的平均度值减小了，从而减小了编译码的代价。LT 码的平均复杂度为 $O(k \log k)$ ，是信息符号数 k 的非线性函数。Raptor 码使用弱化的度分布，每解出一个符号使用 $O(\log(1/\varepsilon))$ 次操作，解出 k 个信息符号总共需要 $O(k \log(1/\varepsilon))$ 次操作，其中 ε 是译码开销，该复杂度是 k 的线性函数。同时 Raptor 码又利用外码的纠错性能保证了误码率。因此，Raptor 码能在无损误码性能的条件下，提高译码器的效率。

2.3 LT 码的编译码原理

LT 码是第一类喷泉码，继承了喷泉码的优点，具有随机线性喷泉码的良好性能，也明显减少了编译码的复杂度。假设信息长度为 k ，平均经过 $O(\ln(k/\delta))$ 次运算生成一个编码符号，每个编码符号都是独立的，平均经过 $O(k \cdot \ln(k/\delta))$ 次异或操作就能从任意的 $k + O(\sqrt{k} \ln^2(k/\delta))$ 个编码符号中，以概率 $1-\delta$ 恢复出 k 个原始信息符号。

2.3.1 LT 码的编码

LT 码的每个编码符号的值都是由所有与之相连的输入符号逐个异或得到的。假设输入 k 个信息符号 s_1, s_2, \dots, s_k ，生成 m 个编码符号 e_1, e_2, \dots, e_m 。LT 码的编码过程如下图 2.4 所示。

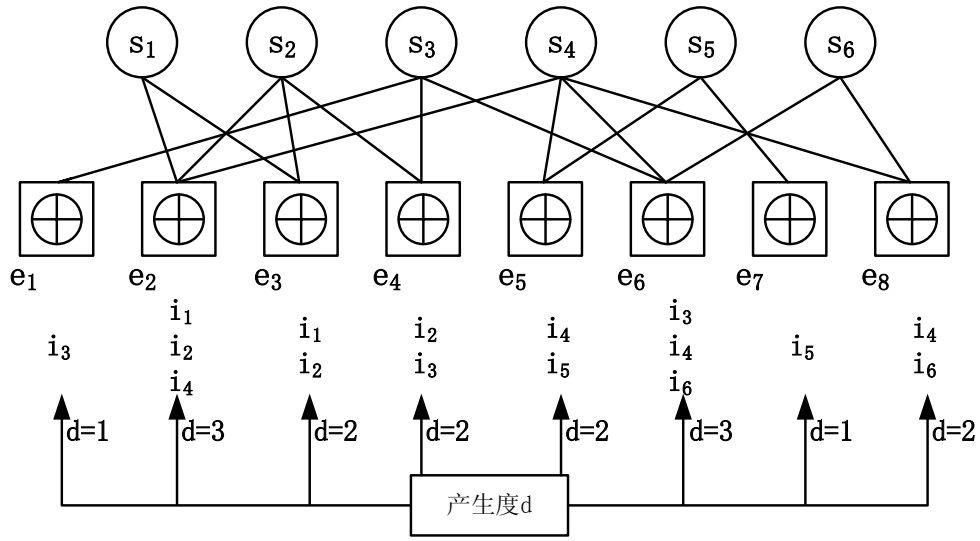


图 2.4 LT 码的编码示意图

上图用双向图表示 LT 码的编码过程。双向图的一边是 k 个输入信息符号，一边是生成的编码符号，每个编码符号都与若干个信息符号相连。每生成一个编码符号，都要随机产生一个度分布 d ，编码器通过如下四步完成编码：

- (1) 根据设计的度数分布 Ω 随机选取一个度数 d ；
- (2) 从 k 个原始信息符号中等概率地随机选取 d 个符号；
- (3) 将这 d 个原始符号逐个异或，生成一个编码符号；
- (4) 重复(1)、(2)、(3)直到编码完成。

LT 码的上述编码过程可用分组码的生成矩阵方法表述：令 X 表示输入符号集， v^i 表示第 i 个编码符号的编码系数向量，LT 码的生成矩阵为

$$G(1, 2, \dots, k) = [v^1 \ v^2 \ \dots \ v^k]^T \quad \text{式(2-4)}$$

那么，LT 编码符号集就是输入符号集和生成矩阵的线性运算，表示为

$$E = [E_1^T \ E_2^T \ \dots \ E_k^T]^T = G(1, 2, \dots, k) \cdot X \quad \text{式(2-5)}$$

其中， E 表示生成的编码符号集。

上述的对 LT 编码过程的两种描述方法分别对应下面将要介绍的两种译码方法。

生成的编码符号通过删除信道不断地被传送出去，接收端要么收到正确的编码符号，要么什么也没有收到。当接收端收到足够多的编码符号时，向发送端发送一个指示，发送端停止编码。接收端译码时，必须知道每一个接收到的编码符号的度和与其相连的输入符号集合，才能译码。实际中，有显式传递方法和隐式传递方法两种解决办法。显式传递就是在编码包中加上编码头部，用来指示编码符号和信息符号之间的双向关系；隐式传递采用一定的算法进行传输，例如基于

抛物线的混沌映射算法^[23]等，节省了编码头部的开销，传输效率更高。

2.3.2 LT 码的译码

BEC 下的 LT 码译码方法很简单，主要有高斯消去(Gaussian elimination)译码算法和置信度传播(BP)译码算法。

高斯消去法的实质就是求解线性方程组式(2- 5)中的 X 。由式(2- 5)可知，只要编码生成矩阵 G 的秩大于 k ，即可成功译码。但是由于高斯消去法需要用大量的时间和空间求矩阵的逆，故这种译码方法只适用于码长较短的情况。

对删除信道上的 LT 码来说，BP 译码算法非常简单。接收端收到多于 k 个的编码符号，就可通过相应的双向图，恢复出输入信息符号。下面我们以图 2.5 的简单的解码过程为例，来说明 BP 译码算法的解码过程。图中， s_k 是待恢复的原始信息符号， t_n 是接收到的编码符号。

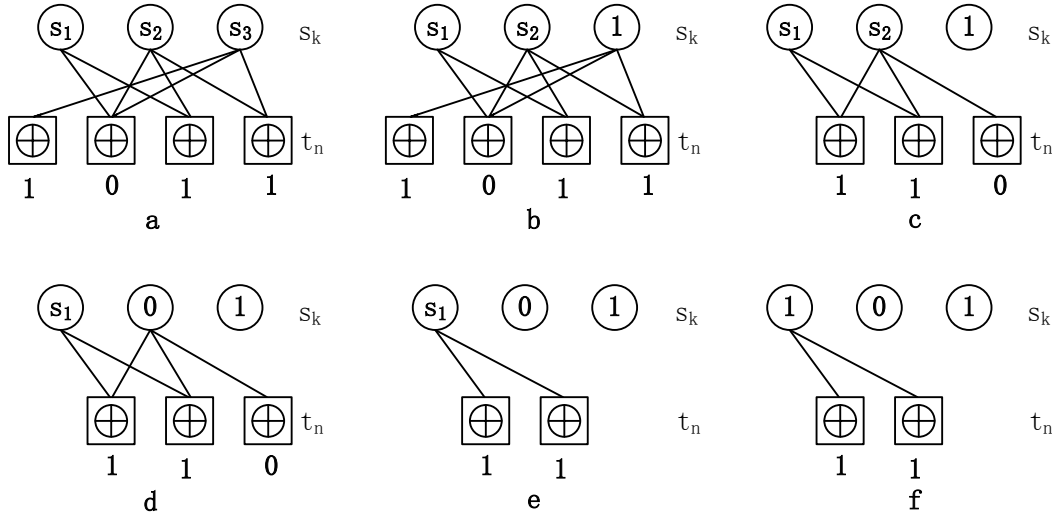


图 2.5 LT 码译码的简单例子

(1)接收一定数量的编码符号，根据双向图在接收到的编码包中寻找度数为 1 的符号 t_n ，即寻找一个编码符号只与一个原始符号 s_k 相连的数据符号。如果这样的 t_n 存在，继续(2)；否则，译码结束；

(2)令 $s_k = t_n$ (如上图 2. 5b)，将与 s_k 相连的所有编码符号与 s_k 相异或，并且相应的度数减 1 (如上图 2. 5c)；

(3)重复(1)、(2)，直到译出所有的 s_k ，或者找不到度数为 1 的编码符号。

译码结束时，还没有被译出的原始信息符号组成的集合称为停止集。相应的，已经被译出的信息符号的集合称为已处理集。

2.3.3 度分布

从 LT 码的编码可以知道，度分布是影响 LT 码译码效率的关键因素。一个好的度分布，一方面应该使平均度数较小，这样才可以减少生成每个编码符号需

要的运算量；另一方面应该给予较大的度数一定选取概率，这样才可以使得 $m \approx k$ 个编码符号覆盖所有的原始信息符号。一般情况下， m 略大于 k ，从而定义译码开销 $\gamma = m/k$ 。

M.Luby 给出了在理论上最理想的度分布——Soliton 分布。该度的分布函数如下：

$$\rho(d) = \begin{cases} \frac{1}{k} & d = 1 \\ \frac{1}{d \cdot (d-1)} & d = 2, 3, \dots, k \end{cases} \quad \text{式(2-6)}$$

孤波分布是根据编码符号恢复信息所需要的期望设计的，也就是在理想状态下能使得每次迭代解码过程中，只有一个校验节点的度数为 1。但围绕理想状态的波动很容易在解码过程中造成在某些点没有度为 1 的编码符号出现，而且更有甚者，一些原始信息符号可能根本就没参与编码，所以这种度分布函数在实际运用中很少采用。

Robust Soliton 分布，又叫鲁棒孤波分布，是在孤波分布的基础上增加了两个参量 c 和 δ 而构造的稳健的度分布。鲁棒孤波分布可以保证在每一次解码中度为 1 的编码符号的数目为 $R = c \ln(k/\delta) \sqrt{k}$ ，而不是 1 个。 δ 是接收到 M 个确知的数据包 t_n 后无法译码的概率的极限， c 是一个自由变量。图 2.6 所示是 $k = 10000$ 时在 δ 一定的条件下， R 与 c 的关系曲线。图中， c 确定时， δ 越大 R 越小；相反， δ 确定时， R 随 c 的增大而增大。

定义这样一个函数：

$$\tau(d) = \begin{cases} \frac{R}{k \cdot d} & d = 1, 2, 3, \dots, \frac{k}{R} - 1 \\ \frac{R}{k} \ln(R/\delta) & d = \frac{k}{R} \\ 0 & d = \frac{k}{R} + 1, \dots, k \end{cases} \quad \text{式(2-7)}$$

再将理想度概率分布 $\rho(d)$ 与 $\tau(d)$ 相加，得到改进后的度概率密度分布函数为

$$\mu(d) = \frac{\rho(d) + \tau(d)}{\beta}, \quad d = 1, 2, \dots, k \quad \text{式(2-8)}$$

其中， $\beta = \sum_{i=1}^k \rho(i) + \tau(i)$ 。

式(2-7)中， $d > K/R$ 时，令 $\tau = 0$ 是为了保证每个原始信息符号都参与编码。理论上，当接收端接收到 $m' = k + 2 \log_e(R/\delta)R$ 个编码符号时，能保证至少以概率

$1-\delta$ 无差错的恢复出原始信息符号。

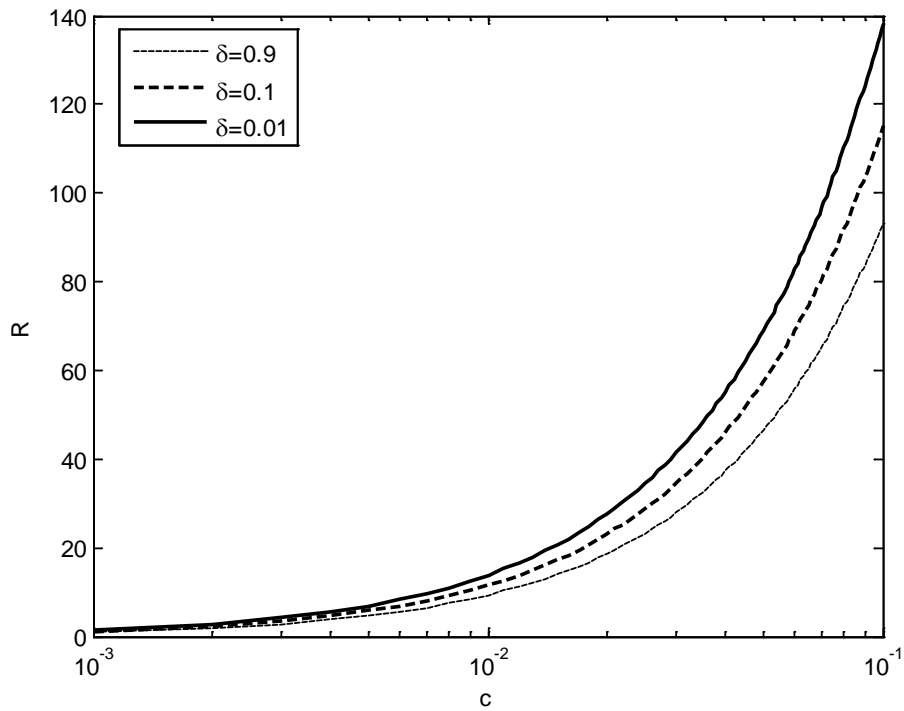


图 2.6 $k=10000$ ，一次迭代中度数为 1 的符号个数 R 与 c 的关系

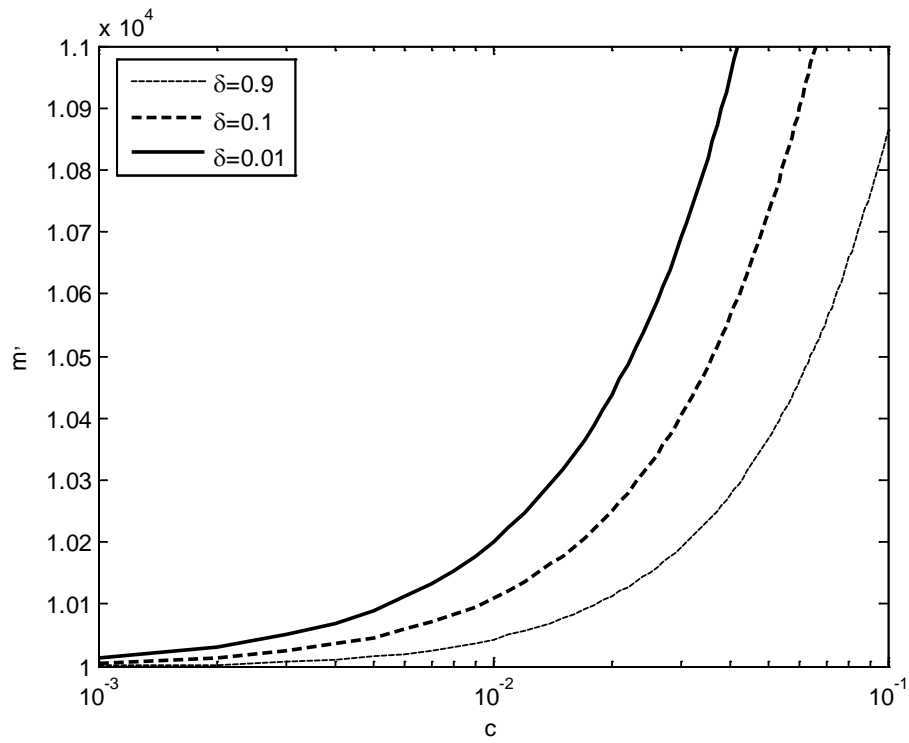


图 2.7 $k=10000$ ，译码需要的编码符号个数 m' 与 c 的关系

图 2.7 所示是 m' 在 δ 一定的条件下随 c 变化的曲线。可以看到， c 越大，需

要接收的编码符号越多，故 c 要尽量小。但是由图 2.6 知， c 越小，一次迭代能译出的符号就越少。因此，要选取合适的 c 才能使译码开销和译码迭代次数都尽量小。

除了上述两种度分布外，文献[8]在构造 Raptor 码时，根据 Robust Soliton 分布将度分布弱化，设计了一种弱化的度分布，分布函数如下：

$$\Omega(x) = 0.00797x + 0.49357x^2 + 0.16622x^3 + 0.07265x^4 + 0.08255x^5 + 0.05606x^8 + 0.03723x^9 + 0.05559x^{19} + 0.02502x^{65} + 0.00314x^{66} \quad \text{式(2-9)}$$

这种分布简单，编码符号的平均度值较小，在 Raptor 码中通常被用来作为 LT 码的度分布。研究 LT 码的非等差错保护时，文献中大多使用的就是这种弱化的度分布，这样能够方便地在 Raptor 码的相关应用中使用。我们在仿真中不作明确说明的，均使用这种弱化的度分布。

2.4 LT 码性能仿真

本节我们将在删除信道下对 LT 码采用 BP 译码算法进行仿真和分析。首先，介绍 LT 码的编译码实现方法，然后给出 LT 码在不同度分布下 LT 码的误比特性能，验证度分布设计对 LT 码性能的影响，以及不同信息符号数对误码率的影响。

2.4.1 LT 码的编译码实现

设信息符号数为 k ，每个符号的长度为 1，采用鲁棒孤波分布。LT 编码时，随机产生长度为 k 的信息序列，然后不断重复如下步骤，直到生成 m 个编码符号：根据度分布随机产生一个度 d ，在信息序列中随机选中 d 个符号，逐个异或，得到一个编码符号，并记录本次编码的生成矩阵 E_l^T ， $l=1,2,\dots,m$ 。在上述过程中，采用下面的方法，实现随机产生一个度值：先计算每个度值的对应的概率，按照各度值对应概率将 $0 \sim k$ 分成若干不等的区间，如图 2.8 所示；然后在 $0 \sim k$ 之间等概随机产生一个数；若该数落在第 $i(i=1,2,\dots)$ 个区间，那么选中的度值就是 i 。

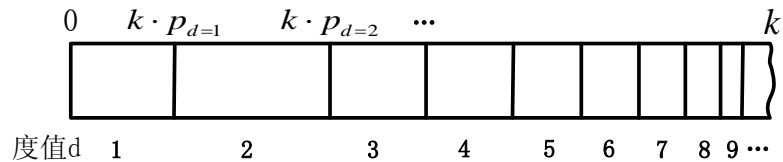


图 2.8 按照度分布选择度值方法示意图

上图中，若选中的随机数为 x ，依次判断 x 所属区间序号。当随机数 x 满足 $k \cdot p_{d=i-1} < x \leq k \cdot p_{d=i}$ ， $i=1,2,\dots$ 时，即 x 落于第 i 个区间，那么本次选中的 $d=i$ 。

根据图 2.5 所示的简单例子，我们给出 LT 译码的实现过程，如图 2.9 所示。

各编码符号的生成矩阵为 E_l^T , $l=1,2,\dots,m$, 所有编码符号对应的生成矩阵为 $G=[E_1^T, E_2^T, \dots, E_m^T]$ 。在 G 中寻找度数为 1 的列, 如图 2.9(a) 的第 1 列, 1 所在的行行为 3, 故令原始信息 $s_3 = t_i|_{i=1}$, 其中 $t_i (i=1,2,\dots,m)$ 是接收到的编码符号, 然后将 G 中第 3 行中的所有非零值都减去 1, 如图 2.9(b) 所示。不断重复上面的步骤, 直到生成矩阵中找不到度数为 1 的列。

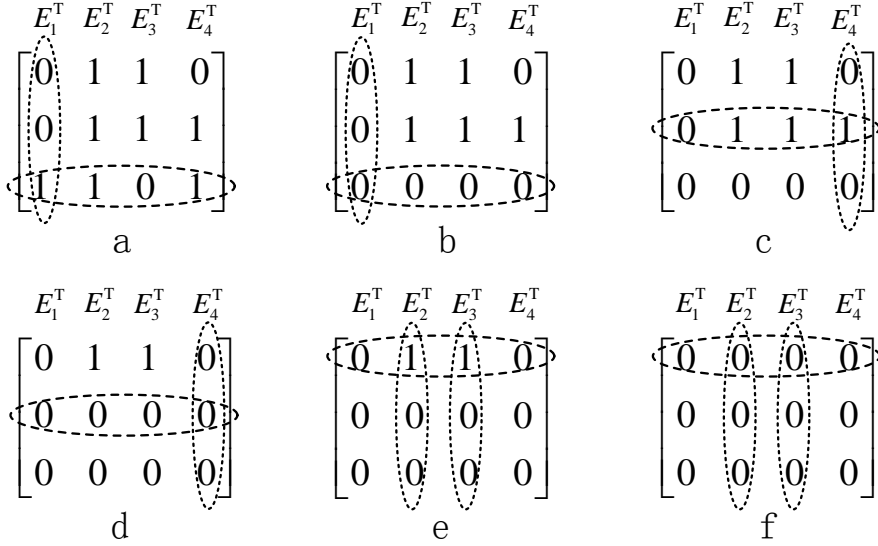


图 2.9 LT 译码实现过程的简单例子

2.4.2 LT 码在 BEC 信道上的性能

喷泉码的性能主要通过译码是否有效来衡量的, 而译码的无效性是由译码过程中存在的失败率引起的, 与 LT 编码过程中的随机行为有关。编码中使用的度分布无疑是影响 LT 码性能的一个因素。当选定度值后, 要在所有的信息符号中随机选取符号参与编码, 因此信息符号的长度也是影响译码性能的因素。下面我们分别使用不同的度分布和不同的信息长度对 LT 码进行仿真。

假设输入的信息符号数为 2000, 鲁棒孤波分布采用的自由变量 $c=0.1$, 译码失败极限为 $\delta=0.5$ 。首先在不同的度分布条件下分别仿真, 验证不同度分布对 LT 码译码性能的影响。

图 2.10 所示是不同度分布下的 LT 码的译码失败率曲线。这里译码失败率定义为: 译码迭代停止时, 停止集中的所有信息符号的个数与总的信息符号数的比值。图中, LT 码在鲁棒孤波分布下的译码失败率明显低于弱化的度分布下的译码失败率。这是因为弱化的度分布放宽了度分布的要求, 导致一些编码符号没有参与编码的概率增大, 即译码停止集增大。在理想的度分布下仿真时, LT 码译码几乎不能进行, 因为迭代中总是找不到度数为 1 的校验节点, 因而 LT 码在理想度分布的译码失败率相当高。后文中没有使用这种度分布, 故这里不给出它的译码

失败率曲线。

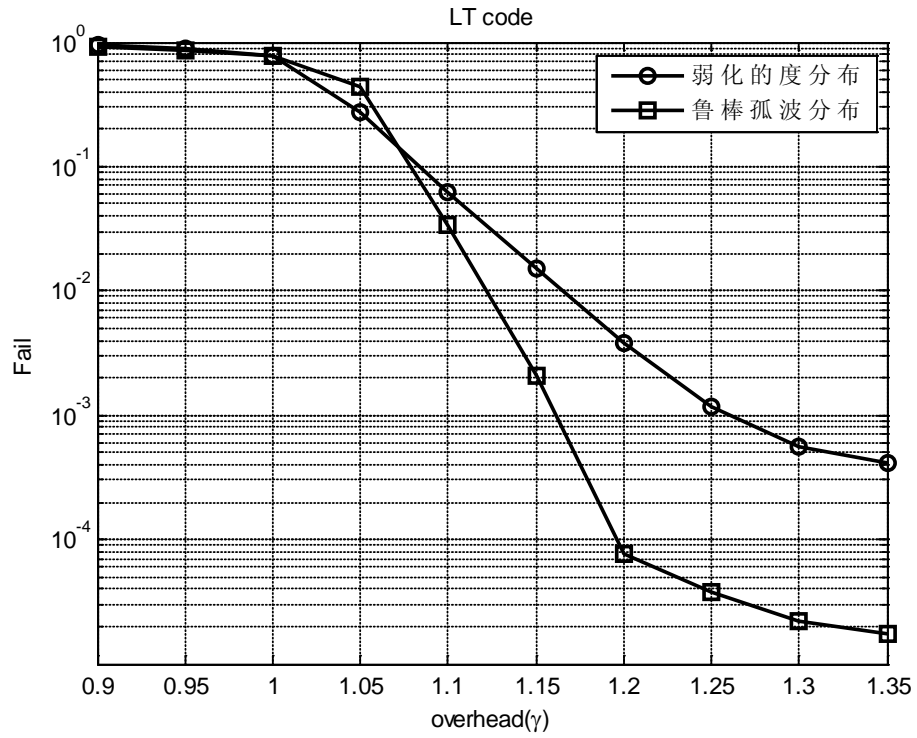


图 2.10 LT 码在不同度分布下的失败率

下面我们验证信息符号长度对译码性能的影响。

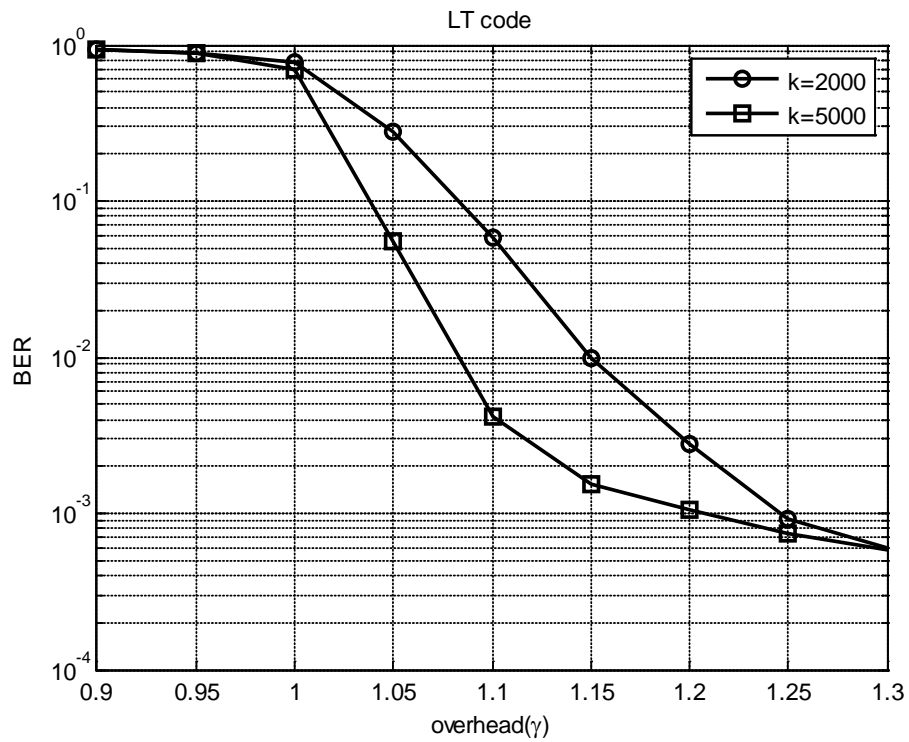


图 2.11 LT 码在不同信息符号长度下的误码率

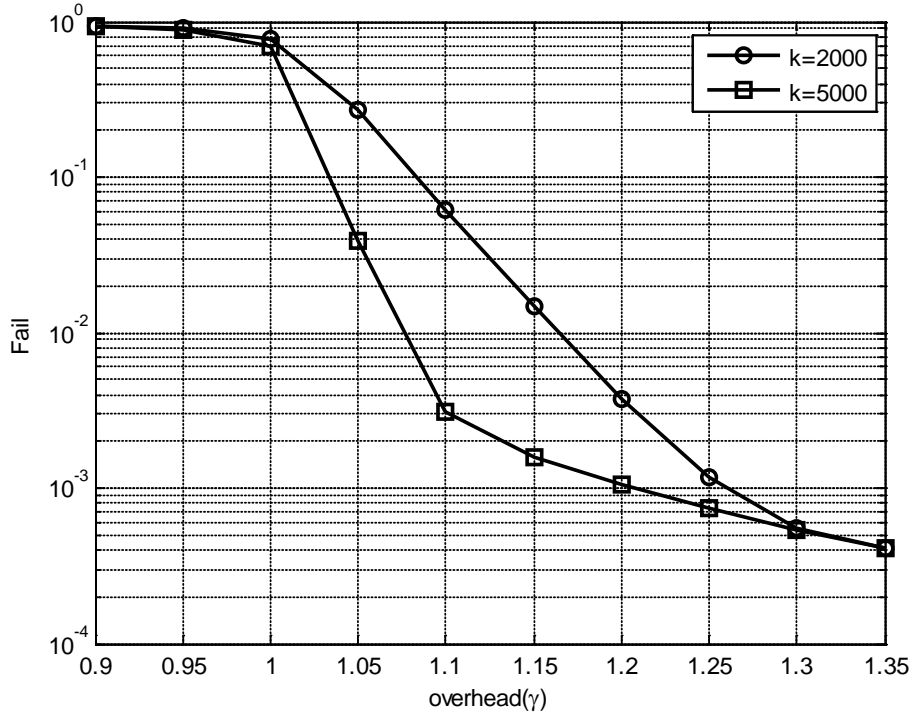


图 2.12 LT 码在不同信息符号长度下的译码失败率

图 2.11 所示是采用弱化的度分布 LT 码在不同输入信息符号长度下的误码率曲线。图 2.12 所示是 LT 码在不同的输入符号长度下的译码失败率曲线。两者的趋势相同,下面以译码失败率为例进行说明。当译码开销较小时,信息符号越长,译码失败率越小;在译码开销较大时,信息符号长度对译码失败率的影响逐渐减小。例如, $\gamma=1.1$ 时,信息符号长度为 2000 时,译码失败率为 $y=6.212 \times 10^{-2}$,而信息长度为 5000 时,译码失败率为 $y=3.131 \times 10^{-3}$;在 $\gamma=1.25$ 处,信息符号长度为 2000 和 5000 时的译码失败率分别为 $y=1.166 \times 10^{-3}$ 和 $y=7.393 \times 10^{-4}$,两者差异逐渐减小。当译码失败率为 0.1 时,信息符号越长,成功译码所需的开销越小。所以 LT 码对码长较长的情况也是适用的。

2.5 LT 码的非等差错保护

随着喷泉码研究的深入,基于喷泉码非等差错保护(UEP)的应用研究也越来越多。UEP方法被广泛应用于视频、音频编码中^[24],例如JPEG2000^[25]、三维SPIHT^[26]等。UEP方法的设计目标在于通过对每个重要等级数据的不同保护,使得在相同的传输条件下,具有较高等级的数据与其他数据相比,在接收端能以较小的误码率被正确恢复出来。通常为实现这一目标,不同重要等级的数据被分配不同的编码冗余度。对于LT码来说,就是为较高等级的信息符号分配较大的选取

概率。

文献中关于 LT 码的 UEP 编码方法主要有三种 UEP-LT 编码方法、EWF 编码方法和重复信息块编码方法，对这些方法的分析通常采用与或树分析法。与或树分析方法（And-Or tree）是 Luby 等人利用先进的概率分析工具，对译码过程的双向图进行研究的分析方法。我们利用该分析方法可以得到基于 LT 码的各种 UEP 方法的理论性能。下面先对与或树分析法做一简单介绍，再以重复信息块编码方法为例进一步说明这种分析方法。UEP-LT 编码方法和 EWF 编码方法将在后面的章节进行详细介绍。

2.5.1 与或树分析法

Luby 等人将随机编码的输入符号看成 or 节点，编码节点看成 and 节点，编码时形成的双向图作为一个 And-Or 树，对其译码过程进行了与或树分析，得到了第 l 次译码的理论误码率。

将 n 个输入信息符号分为 r 个集合 S_1, S_2, \dots, S_r ，每个等级的符号数为 α_i ($i=1, 2, \dots, r$)，显然有 $\sum_i \alpha_i = 1$ 。编码时采用的度分布函数为 $\Omega(x)$ ， $\mu = \Omega'(1)$ 为编码的平均度数。令 q_i 表示双向图中一条边与 S_i 集合相连的概率。若接收端收到的符号数为 m ，译码开销 $\gamma = m/n$ 。

文献[27]给出了第 j 个优先级在第 l 次迭代译码后误码率为：

$$\begin{aligned} y_{0,j} &= 1 \\ y_{l,j} &= \delta_j \left(1 - \beta \left(1 - \sum_{k=1}^r q_k y_{l-1,k} \right) \right), \quad l \geq 1 \end{aligned} \quad \text{式(2-10)}$$

$$\text{其中, } \beta(x) = \frac{\Omega'(x)}{\Omega'(1)}, \quad \delta_j(x) = e^{np_j \mu \gamma (x-1)}$$

这里， p_j 表示双向图中一条边与集合 S_j 中一个输入符号相连的概率。定义

$$G_{l,i,j} = \frac{y_{l,i}}{y_{l,j}}, \quad \text{容易看到, } G_{l,i,j} \text{ 越小, 集合 } S_i \text{ 中的符号相较于 } S_j \text{ 中的符号的误码率}$$

越低，即被恢复的概率越高。由式(2-10)可得

$$G_{l,i,j} = e^{-n(p_i - p_j) \mu \gamma \beta \left(1 - \sum_{k=1}^r q_k y_{l-1,k} \right)}, \quad l \geq 1 \quad \text{式(2-11)}$$

容易看出，当且仅当 $p_i > p_j$ 时，有 $G_{l,i,j} < 1$ ，集合 S_i 的误码率低于集合 S_j ，即 $y_{l,i} < y_{l,j}$ 。也就是说，要增大某集合在译码时的恢复概率，就要增大该集合中每个符号的被选取概率。因此，在实现 LT 码的非等差错保护时，一种有效的办法就是以较大的概率选取优先级较高的集合中的输入符号。

2.5.2 重复信息块编码方法

将输入的信息符号分成重要比特(More Important Bits, 简称 MIB)和不重要比特(Less Important Bits, 简称 LIB)两个等级。

重复信息块法, 是将信息重复之后得到一个“虚拟”的信息序列, 在度分布中使用该“虚拟”信息序列的长度, 分配各度值的概率。若将 MIB 的信息多重复几次, 实际上就是增大 MIB 集合在总信息中所占的比例, 同时也改变了各度值的分布概率, 从而达到非等差错保护的目的。下面详细介绍, 采用的重复策略以及当序号超出实际信息序号范围时的处理办法。

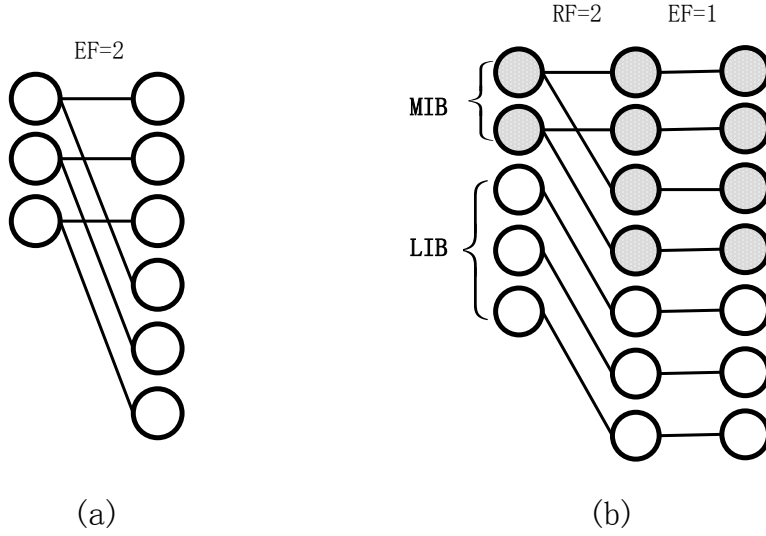


图 2.13 重复信息块编码方法示意图

假设输入信息符号长度为 k , 采用鲁棒孤波分布, 度数集合 Ω_d 为 $\{1, 2, \dots, k\}$ 。如图 2.13(a)所示, 将原始信息按顺序重复 EF 次, 编码时按照度数集合为 $\{1, 2, \dots, EF \times k\}$ 的度分布进行编码。如果选到的符号序号大于 k , 就将该序号减去 ck , 其中 $c=1, 2, \dots, EF-1$, 得到该信息的实际序号。这样编码完成时, 得到的生成矩阵依然是 $m \times k$ 维, 接收端的译码依然可以使用信息长度为 k 的译码。

将上述方法应用于分等级的情况, 假设信息分成两个等级 MIB 集合和 LIB 集合, 如图 2.13(b)所示: 将 MIB 集合中的符号重复 RF 次, 按顺序置于原 MIB 集合之后, 再将全部符号重复 EF 次, 置于全部符号之后。此时, 采用鲁棒孤波分布的度数集合为 $\{1, 2, \dots, (n_{MIB}RF + n_{LIB}) \cdot EF\}$, 其中, n_{MIB} 是 MIB 集合的符号数, n_{LIB} 是 LIB 集合的符号数。当参与编码的信息序号大于 $RFn_{MIB} + n_{LIB}$ 时, 信息序号减去 $c \cdot (RFn_{MIB} + n_{LIB})$, $c=1, 2, \dots, EF-1$, 得到一个新的信息序号, 若该信息序号仍然大于 k , 将它减去 $c' n_{MIB}$, 其中 $c'=1, 2, \dots, (RF-1)n_{MIB}$, 目的是保证信息序号不大于 k , 从而得到该信息的原始序号。

特殊地, $RF = EF = 1$ 时, 重复信息块编码是一般的不分等级的 LT 编码。

因为这种编码方法在编码中要调整度数空间的元素，实际上是根据信息符号的长度改变了各度数的概率分布，故一般采用度数连续的鲁棒孤波分布。对于式(2-9)的与信息符号长度无关的度分布，重复信息块编码方法很难发挥其作用。

下面我们对这种方法进行与或树分析，期望得到它在理论上的误码率极限。

当 MIB 集合中符号被重复 RF ，全部信息再重复 EF 次后，MIB 集合的符号数由原来的 αn 变成 $R_M \alpha n$ ，其中

$$\begin{aligned} R_M &= \frac{EF \cdot RF}{EF \cdot RF \cdot \alpha + EF \cdot (1 - \alpha)} \\ &= \frac{RF}{1 - \alpha + \alpha RF} \end{aligned} \quad \text{式(2-12)}$$

易知，MIB 集合的被选中概率 $q_1 = \alpha R_M$ ，MIB 集合中每个符号被选取的概率

为 $p_1 = \frac{\alpha R_M}{\alpha n} = \frac{R_M}{n}$ 。相应的，对于 LIB 集合来说， $q_2 = 1 - \alpha R_M$ ， $p_2 = \frac{1 - \alpha R_M}{(1 - \alpha)n}$ 。

由此根据式(2-10)得到，MIB 集合和 LIB 集合的误码率公式：

$$\begin{aligned} y_{0,M} &= y_{0,L} = 1 \\ y_{l,M} &= e^{-n\gamma \mu \frac{R_M}{n} \beta (1 - \alpha R_M y_{l-1,M} - (1 - \alpha R_M) y_{l-1,L})} \\ y_{l,L} &= e^{-n\gamma \mu \frac{1 - \alpha R_M}{(1 - \alpha)n} \beta (1 - \alpha R_M y_{l-1,M} - (1 - \alpha R_M) y_{l-1,L})} \end{aligned} \quad \text{式(2-13)}$$

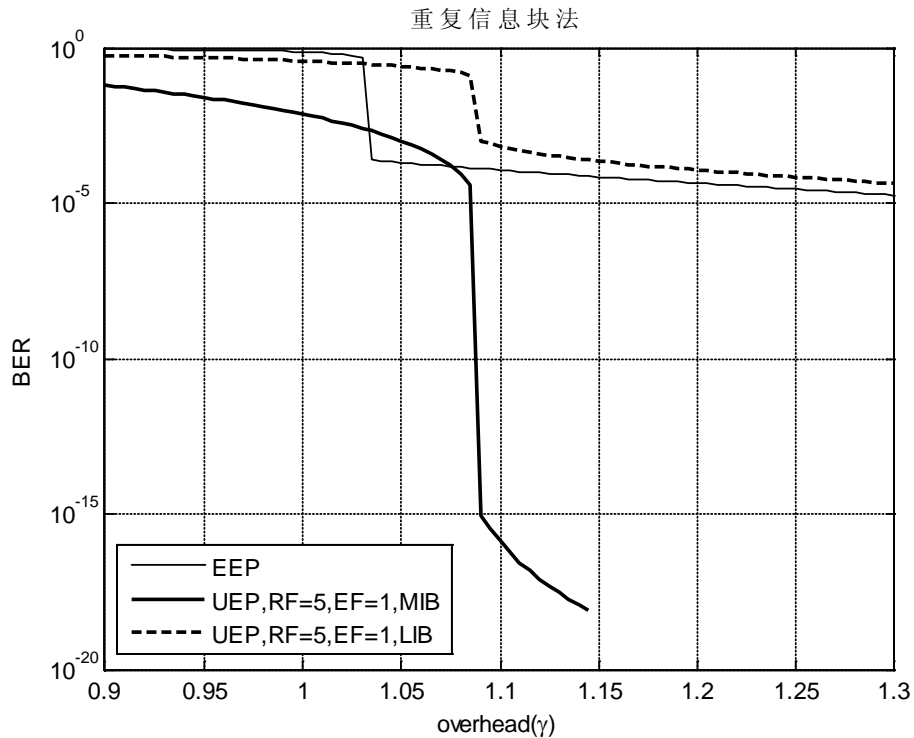


图 2.14 重复信息块编码方法的误码率随译码开销变化的渐近线

若信息符号长度为 2000，其中有 100 个 MIB 集合的符号。根据上式(2-13)，得到重复信息块编码方法的误码率极限如图 2.14 所示。图中采用的鲁棒孤波分布的参数分别为 $c = 0.1$ ， $\delta = 0.5$ 。当译码开销 γ 为 1.09 时，MIB 集合的误码率骤降至 9.233×10^{-16} 。当 $\gamma < 1.085$ 时，误码率曲线如图 2.15 所示。在 $\gamma = 1.085$ 处，误码率为 3.835×10^{-5} 。

图中当译码开销趋近于 0 时，不分等级编码的误码率趋近于 1。仿真时假设信息是随机产生的 0、1 序列，译码的初始序列是全-1 序列，当迭代没有进行时信息序列全错，即误码率的初始值是 1。

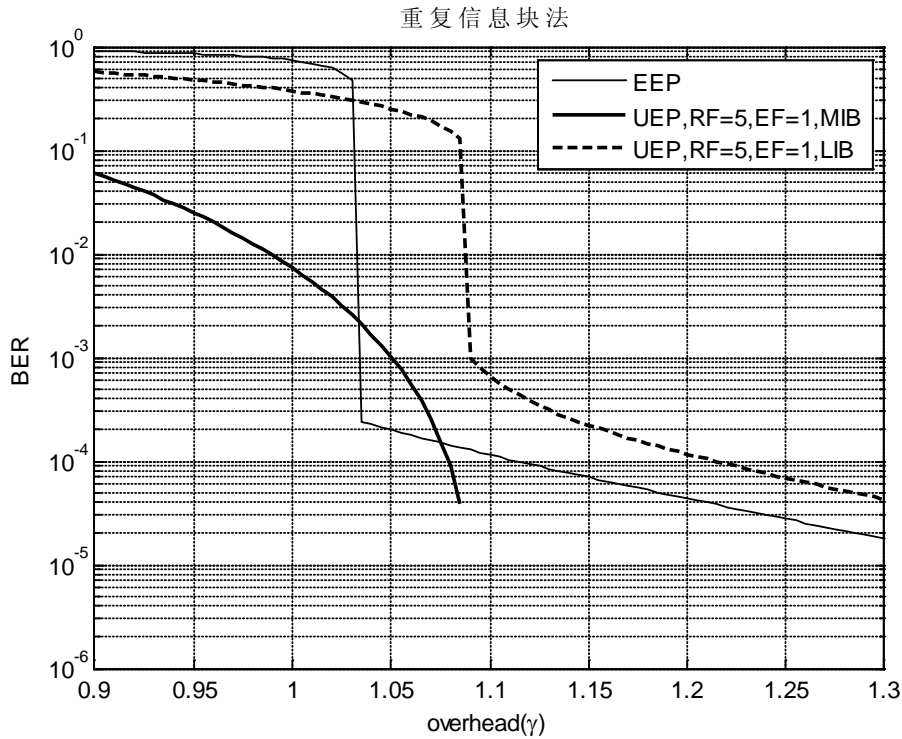


图 2.15 $\gamma < 1.085$ 时，重复信息块编码方法的误码率随译码开销变化的渐近线

我们在如下条件下，对重复信息块编码方法进行仿真：输入 2000 个信息符号，分成 MIB 和 LIB 两个集合，其中 MIB 集合的符号占 0.05；度分布采用鲁棒孤波分布，其中 $c = 0.1$ ， $\delta = 0.5$ 。仿真中，选取的参数 $RF = 5$ ， $EF = 1$ ，仿真结果如图 2.16 所示。

图中，当 $\gamma = 1.15$ 时，MIB 集合的误码率为 $y_M = 4.325 \times 10^{-7}$ ，LIB 集合的误码率为 $y_L = 4.863 \times 10^{-3}$ 。但是因为重复信息块的方法要根据“虚拟”的符号长度调整度分布函数，故使用的度分布是与信息符号有关的鲁棒孤波分布。前文中提到的弱化的度分布与符号长度无关，不能用在重复信息块编码方法中。所以对于重复信息块编码方法来说，限制了它的应用扩展。

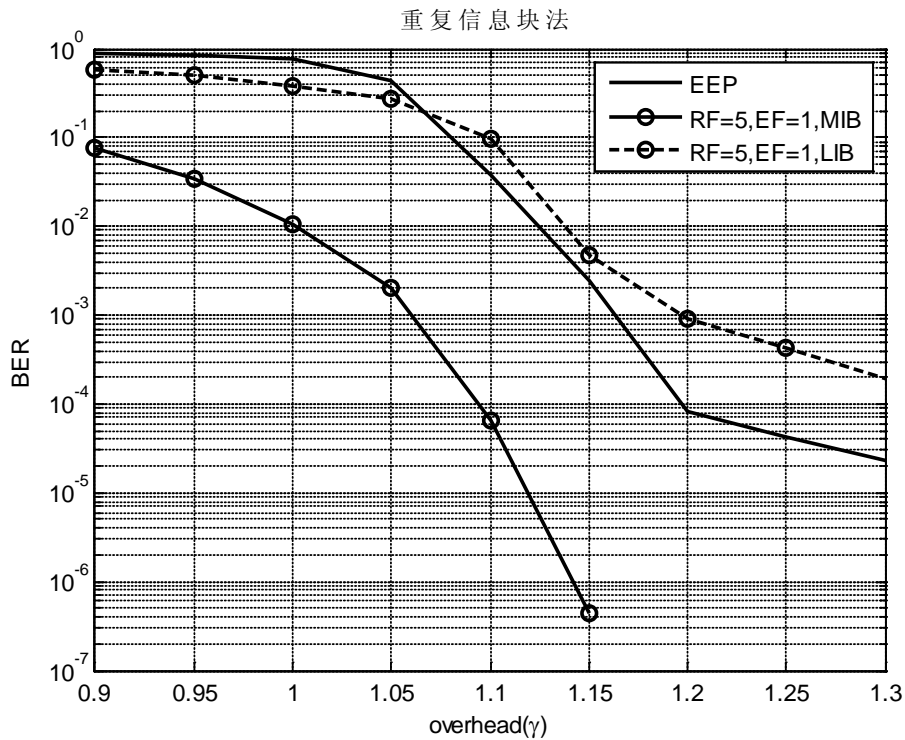


图 2.16 重复信息块编码方法的误码率仿真曲线

2.6 本章小结

本章主要介绍了第一类喷泉码——LT 码的相关内容，包括 LT 码的编译码原理、性能仿真及 UEP 方法的理论分析方法等。

本章首先在详细介绍 LT 码的编译码原理及其具体实现方法的基础上，对 LT 码在 BEC 信道上的性能进行了大量仿真，验证了度分布及输入信息符号的长度对 LT 码性能的影响。仿真证明，鲁棒孤波分布下的 LT 码性能最优，但是它的平均度值较大，编码的复杂度较弱化的度分布大。在弱化的度分布下，输入信息符号长度越长成功译码所需要的开销越低，说明 LT 码适用于码长 K 较大情况的编译码。

本章最后介绍了一种适用于 LT 码的非等差错保护分析与或树分析方法，并以重复信息块编码方法为例，进一步说明了这种分析方法。本文在后面的章节中均采用这种与或树分析方法对 LT 码的各种非等差错保护方法进行分析。

第三章 基于扩展窗的逐个选择法

本章提出了一种新的 LT 码的非等差错保护方法——基于扩展窗的逐个选择法。该方法给出了一种新的符号选取方法，能够使得重要等级的符号选取概率大于不重要等级的符号选取概率，达到非等差错保护的目的。受 EWF 编码方法的启发，本章提出的方法用两个窗覆盖所有的输入信息符号，选择每个参与编码的信息符号前要先选择一个窗。

本章首先介绍 EWF 编码方法，再给出基于扩展窗的逐个选择法的编码方法，最后对这两种编码方法进行对比分析，说明本章提出方法的误码性能。

为方便下面的介绍，我们规定以下几个符号的意义。LT 编码时，度分布函数为 $\Omega(x) = \sum_{i=1}^n \Omega_i x^i$ ，编码的平均度值为 $\mu = \Omega'(1)$ 。 n 个信息分为 r 个集合 S_1, S_2, \dots, S_r ，每个集合的符号数为 $\alpha_i n$ ，($i = 1, 2, \dots, r$)。接收端收到的编码符号为 m 个，译码开销(overhead) $\gamma = m/n$ 。分析及仿真均以两个等级为例，将输入的信息符号分成重要比特(More Important Bits, 简称 MIB)和不重要比特(Less Important Bits, 简称 LIB)两个等级。

3.1 EWF 编码方法介绍

EWF 编码方法是 D. Sejdinovic 等^[19]提出的一种性能良好的 LT 码的 UEP 编码方法。该方法将编码符号分成两类，一类只有 MIB 集合的符号参与编码，一类由 MIB 集合和 LIB 集合共同参与编码。显然，MIB 集合总有机会参与编码，故 MIB 集合中符号的选取概率总是大于等于 LIB 集合。这样，就可以达到非等差错保护的目的。

EWF 编码方法是用窗 W_i , ($i = 1, 2, \dots, r$) 覆盖所有的输入信息符号，窗 W_i 包含 S_1, S_2, \dots, S_i 中的所有符号，对应的度分布函数为 $\Omega_i(x)$ 。 $r = 2$ 时，EWF 编码示意图如图 3.1 所示。

图中， i_1, i_2 属于 MIB 集合 S_1 ， $i_3 \sim i_6$ 属于 LIB 集合 S_2 ，每个扩展窗可采用不同的度分布，本文中为每个扩展窗选择相同的度分布。LT 编码时，先以概率 Γ_1 选取窗 W_1 ，以概率 $(1 - \Gamma_1)$ 选取窗 W_2 ，再根据度分布 $\Omega(x)$ 选取一个度数 d ，然后在选到的窗中等概随机选取 d 个符号相异或，得到一个编码符号。这样不断重复生成编码符号，直到编码完成。

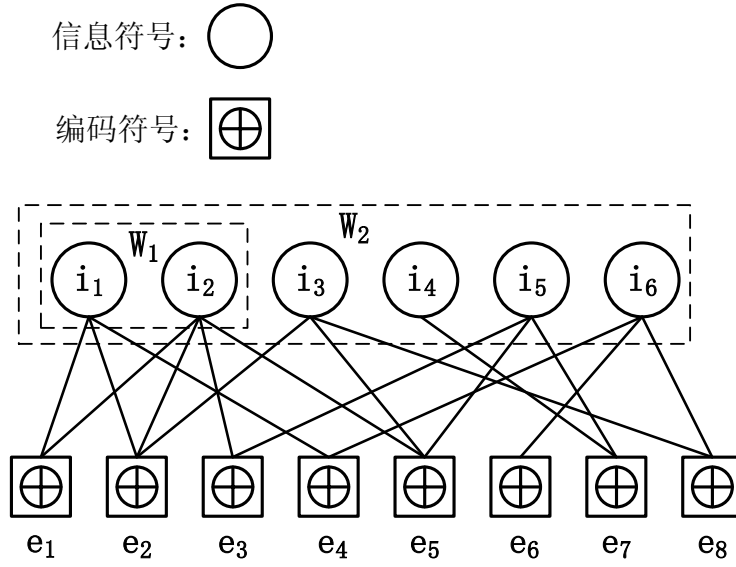


图 3.1 EWF 编码示意图

由上述编码过程可知, **MIB** 集合的选取概率大于 **LIB** 集合, 相应的 **MIB** 集合中每个符号的选取概率也大于 **LIB** 集合中每符号的选取概率。

编码方法在生成某个编码符号时, 所选取的符号要么全部来自窗 W_1 , 要么全部来自窗 W_2 。因此, 对 EWF 编码方法进行与或树分析时, 要对这两种情况分别进行考虑。

当编码选取的符号来自窗 W_1 时, 参与编码的符号不包含 **LIB** 集合符号的信息, 此时 **LIB** 集合中符号被选取的概率 $p_{LIB} = 0$ 。选中窗 W_1 的条件下, **MIB** 集合中每个符号被选取的概率为 $p_{MIB} = \Gamma_1 / \alpha_1 n$, **MIB** 集合被选中的概率 $q_{MIB} = 1$, 故 **MIB** 集合和 **LIB** 集合的误码率分别如下式:

$$\begin{aligned} y_{0,M_1} &= y_{0,L_1} = 1 \\ y_{l,M_1} &= e^{-\frac{\Gamma_1 \Omega (1 - y_{l-1,M})}{\alpha_1}}, l \geq 1 \\ y_{l,L_1} &= 1, l \geq 1 \end{aligned} \quad \text{式(3-1)}$$

若编码选取的符号全部来自窗 W_2 时, **MIB** 集合和 **LIB** 集合中每个符号的选取概率相等为 $p_{MIB} = p_{LIB} = \Gamma_2 / n$, **MIB** 集合以概率 $q_{MIB} = \alpha_1$ 被选取, 相应的 **LIB** 集合的选取概率为 $q_{LIB} = 1 - \alpha_1$, 故 **MIB** 集合和 **LIB** 集合在这种情况下的误码率为:

$$\begin{aligned} y_{0,M_2} &= y_{0,L_2} = 1 \\ y_{l,M_2} &= y_{l,L_2} = e^{-\Gamma_2 \Omega (1 - \alpha_1 y_{l-1,M} - (1 - \alpha_1) y_{l-1,L})}, l \geq 1 \end{aligned} \quad \text{式(3-2)}$$

根据与或树分析, 编码符号是与节点, 两种情况间满足相“与”的关系, 故 EWF 编码方法的 **MIB** 集合和 **LIB** 集合的误码率由式(3-1)、式(3-2)可得

$$\begin{aligned}
y_{0,M} &= y_{0,L} = 1 \\
y_{l,M} &= e^{-\gamma \left(\frac{\Gamma_1}{\alpha} \Omega (1 - y_{l-1,M}) + \Gamma_2 \Omega (1 - \alpha y_{l-1,M} - (1 - \alpha) y_{l-1,L}) \right)}, l \geq 1 \\
y_{l,L} &= e^{-\gamma \Gamma_2 \Omega (1 - \alpha y_{l-1,M} - (1 - \alpha) y_{l-1,L})}, l \geq 1
\end{aligned} \tag{3-3}$$

特别地， $\Gamma_1 = 0$ 时，EWF 编码是一般的不分等级（EEP）编码的情况。

根据式(3-3)可以得到图 3.2、图 3.3 的理论曲线。

图 3.2 是误比特率随着窗 W_1 的选择概率 Γ_1 变化的渐近线。从图中可以看到，译码开销 γ 越大，MIB 集合和 LIB 集合的误码率越小。当译码开销 $\gamma = 1.05$ 时，在 $\Gamma_1 = 0.083$ 处，MIB 集合的误比特率为 $y_{\infty,M} = 5.303 \times 10^{-5}$ ，LIB 集合的误码率为 $y_{\infty,L} = 8.777 \times 10^{-3}$ 。而当 $\Gamma_1 = 0$ ，即等差错保护时，误码率为 $y_{\infty} = 3.402 \times 10^{-3}$ 。说明 MIB 集合的误码性能提升的同时 LIB 集合的误码性能降低。

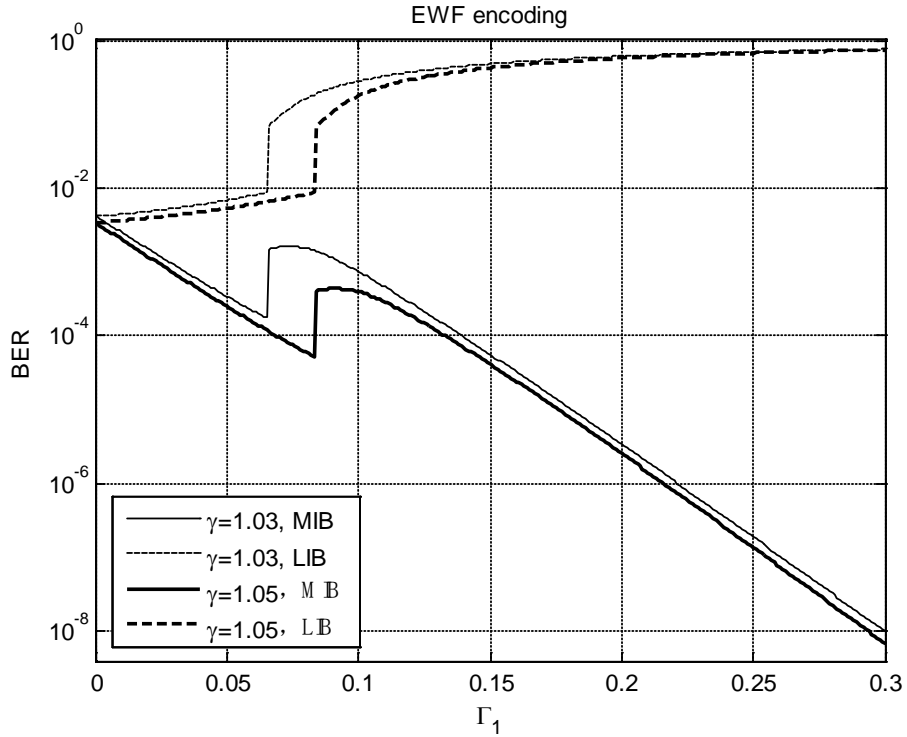


图 3.2 EWF 编码的误码率随 Γ_1 变化的渐近线

图 3.3 是 $\Gamma_1 = 0.084$ 及 $\Gamma_1 = 0.11$ 时误比特率 BER 随译码开销 γ 的变化曲线。图中，译码开销 $\gamma = 1.25$ ， $\Gamma_1 = 0.084$ 时， $y_{\infty,M} = 3.158 \times 10^{-6}$ ， $y_{\infty,L} = 1.5 \times 10^{-3}$ ； $\Gamma_1 = 0.11$ 时， $y_{\infty,M} = 5.953 \times 10^{-7}$ ， $y_{\infty,L} = 1.906 \times 10^{-3}$ 。 Γ_1 越大，MIB 集合的误码率越小，LIB 集合的误码率越大，故参数 Γ_1 要合理选择。当 $\Gamma_1 = 0.084$ 时，误码率陡降时的译码开销在 1.05 附近，比 $\Gamma_1 = 0.11$ 时的小。

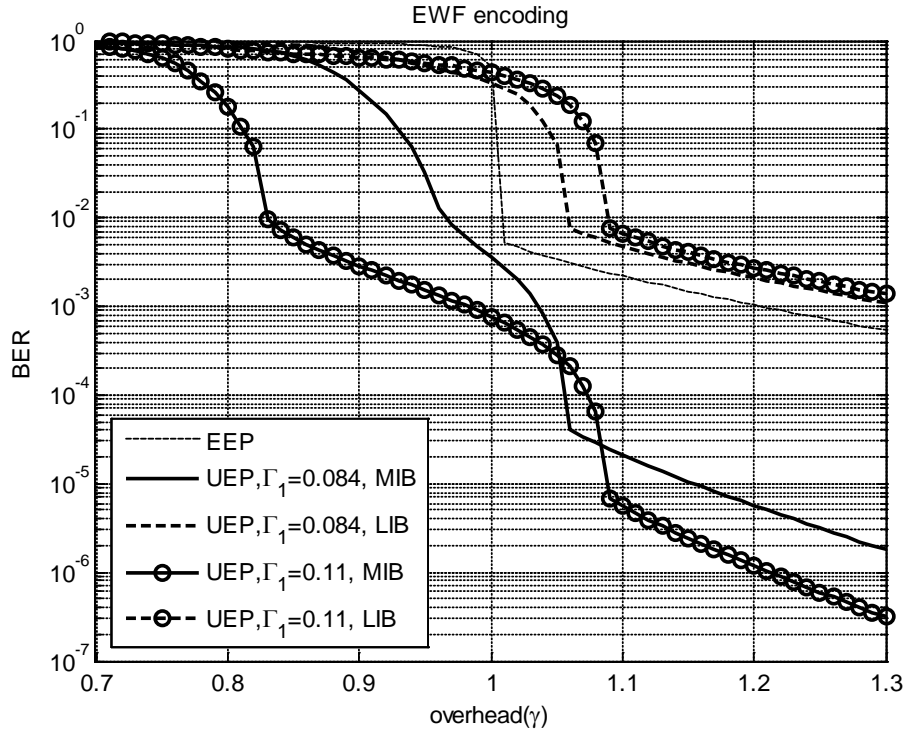


图 3.3 EWF 编码误码率随译码开销变化的渐近线

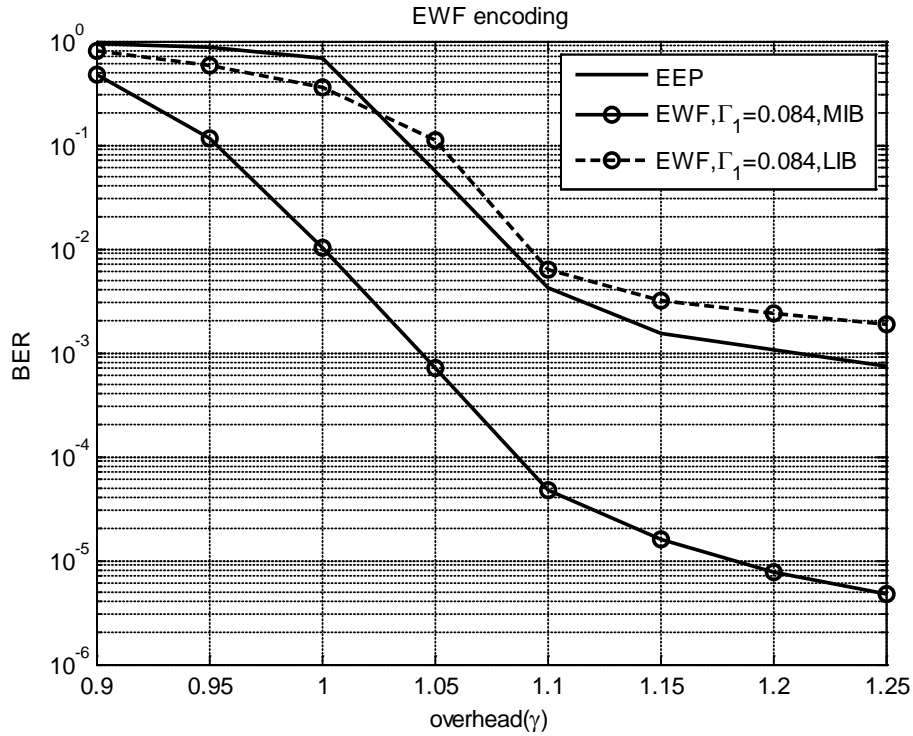


图 3.4 EWF 编码的误码率仿真曲线

假设输入符号数为 5000，MIB 集合的符号数占总符号数的 0.1，选择参数 $\Gamma_1 = 0.084$ 。此时， $\Gamma_2 = 1 - 0.084 = 0.916$ ， $p_1 = \frac{\Gamma_1}{\alpha n} + \frac{\Gamma_2}{n} = 3.512 \times 10^{-4}$ ，

$p_2 = \frac{\Gamma_2}{n} = 1.832 \times 10^{-4}$ 。在 BEC 信道下仿真得到的误码率曲线如图 3.4 所示。图中，当 $\gamma = 1.25$ 时，MIB 集合的误码率为 $y_M = 4.767 \times 10^{-6}$ ，此时 LIB 集合的误码率为 $y_L = 1.903 \times 10^{-3}$ ，比 EEP 的误码率稍差一些。所以 MIB 集合的误码性能的提升以 LIB 集合的误码性能下降为代价，实际应用中要充分考虑两个等级的误码需求。

3.2 基于扩展窗的逐个选择法

基于扩展窗的逐个选择编码方法受 EWF 编码方法启发，考虑用窗 W_1 、 W_2 覆盖所有输入信息符号，但是符号选择方法有所不同。基于扩展窗的逐个选择法的编码时，根据度分布选取一个度数 d ，然后依次选出 d 个符号参与编码。选取每一个符号前，先以概率 Γ_1 选取窗 W_1 ，以概率 Γ_2 选取窗 W_2 ，其中 $\Gamma_2 = 1 - \Gamma_1$ ，在选到的窗中等概随机选取一个符号。这样，先选择窗再选择符号，重复 d 次，将选出的 d 个符号逐个异或，得到一个编码符号。不断重复上面的步骤生成编码符号，直到编码完成。

这种逐个选取参与编码符号的方法，能保证在生成每一个编码符号时，MIB 集合的符号的选取概率均大于 LIB 集合，同时减少了某次编码中只有 MIB 集合的符号信息而没有 LIB 集合的信息的情况。

对于上述的编码过程，MIB 集合的符号选取概率为 $p_1 = \frac{\Gamma_1}{\alpha n} + \frac{\Gamma_2}{n}$ ，LIB 集合的符号选取概率为 $p_2 = \frac{\Gamma_2}{n}$ ，第 i 个集合被选中的概率为 $q_i = p_i \alpha_i n$ ，其中 $i = 1, 2$ ，分别对应集合 MIB 和 LIB。

将 p 、 q 值代入式(2-10)，分别得到 MIB 集合和 LIB 集合的误码率公式：

$$\begin{aligned} y_{0,M} &= y_{0,L} = 1 \\ y_{l,M} &= e^{-\gamma \left(\frac{\Gamma_1}{\alpha} + \Gamma_2 \right) \Omega \left(1 - (\Gamma_1 + \alpha \Gamma_2) y_{l-1,M} - (1 - \alpha) \Gamma_2 y_{l-1,L} \right)}, l \geq 1 \\ y_{l,L} &= e^{-\gamma \Gamma_2 \Omega \left(1 - (\Gamma_1 + \alpha \Gamma_2) y_{l-1,M} - (1 - \alpha) \Gamma_2 y_{l-1,L} \right)}, l \geq 1 \end{aligned} \quad \text{式(3-4)}$$

根据上式(3-4)得到 BER 随 Γ_1 的变化曲线如图 3.5 所示。译码开销 $\gamma = 1.05$ 时， $\Gamma_1 = 0.12$ 处，MIB 集合的误比特率为 $y_{\infty,M} = 3.621 \times 10^{-5}$ ，优于 EWF 编码方法的 MIB 集合在 $\Gamma_1 = 0.084$ 处的 $y_{\infty,M} = 3.158 \times 10^{-6}$ ；LIB 集合的误码率为 $y_{\infty,L} = 1.321 \times 10^{-2}$ ，较 EWF 编码方法差。说明对于 MIB 集合来说，基于扩展窗的逐个选择法的误码率是可以达到比 EWF 编码方法更好的误码性能的。

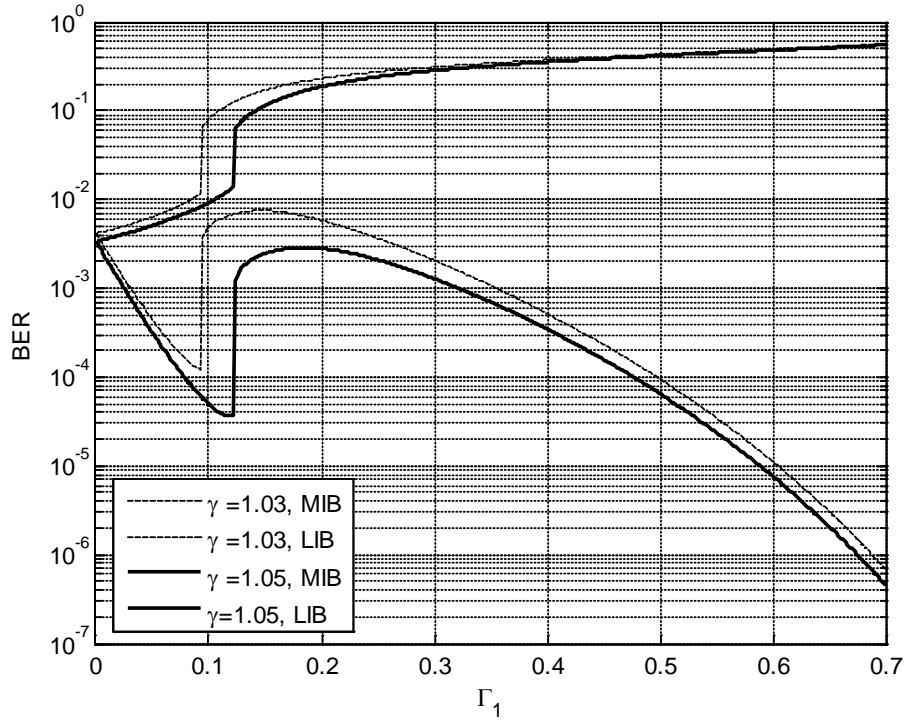


图 3.5 基于扩展窗的逐个选择法的误码率随参数 Γ_1 变化的渐近线

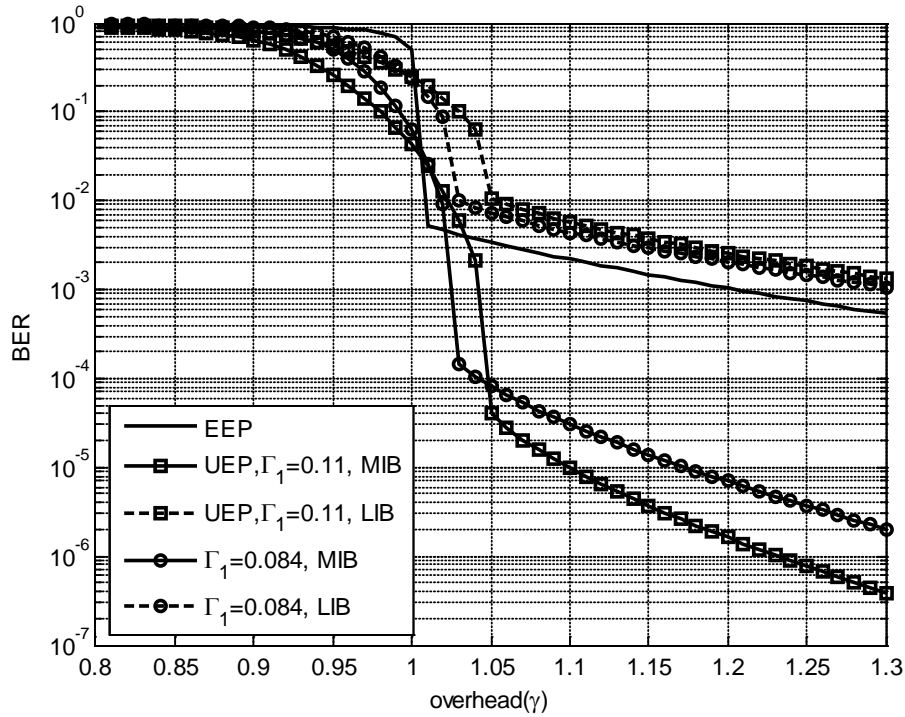


图 3.6 基于扩展窗的逐个选择法的误码率随译码开销变化的渐近线

BER 随译码开销 overhead 变化的渐近线如图 3.6 所示。以 $\Gamma_1 = 0.11$ 为例，

$\gamma=1.25$ 时, MIB 集合的误码率为 $y_{\infty,M}=7.639\times 10^{-7}$, LIB 集合的误码率为 $y_{\infty,L}=1.838\times 10^{-3}$ 。图中, Γ_1 越大 MIB 集合的误码率越小, LIB 集合的误码率越大, 误码率陡然下降时的译码开销越大, 所以 Γ_1 并不是越大越好。

假设输入信息符号 5000, $\alpha_1=0.1$, 基于扩展窗的逐个选择法的仿真曲线如下图 3.7 所示, proposed1 即指该方法。

图中, 对于 MIB 集合来说, 总体上 $\Gamma_1=0.084$ 的误码性能比 $\Gamma_1=0.11$ 时的略差。这是因为 $\Gamma_1=0.084$ 时 MIB 集合的符号选取概率比 $\Gamma_1=0.11$ 时的小。说明在较大译码开销下, Γ_1 越大, MIB 集合的误码率越低, LIB 集合的误码率越高, 即 MIB 集合和 LIB 集合之间的误码差异越明显。但是当译码开销等于 1.05 时, $\Gamma_1=0.084$ 的误码率低于 $\Gamma_1=0.11$ 的误码率, 这与理论曲线是相符的。因为 $\Gamma_1=0.084$ 时误码率陡降要求的译码开销比 $\Gamma_1=0.11$ 时的小, 故误码率能够稍低于后者, 而后者经过陡降误码性能会超越前者。

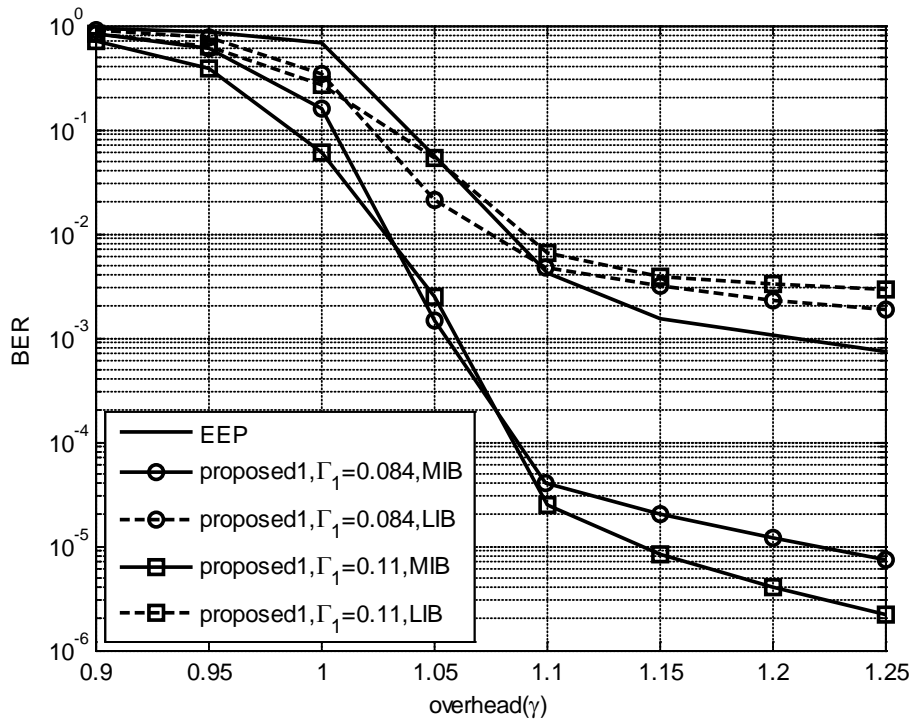


图 3.7 基于扩展窗的逐个选择法在不同参数下的仿真曲线

3.3 基于扩展窗的逐个选择法的仿真对比

本节将从编码复杂度和误码性能两个方面对基于扩展窗的逐个选择法和 EWF 编码方法进行对比。

因为用基于扩展窗的逐个选择法编码时, 选择每个参与编码的符号之前, 都要先选择窗, 所以这种方法与 EWF 方法相比, 编码复杂度有所增加。采用式(2-9)

的弱化的度分布, 平均编码度数 $\mu \approx 5.86$, 平均约增加了 $4.86m$ 次选择, 与后文中介绍的 UEP-LT 编码方法的复杂度相同。

假设信息符号长度为 5000, MIB 集合的符号数为 500, 基于扩展窗的逐个选择法和 EWF 法的仿真结果如图 3.8、图 3.9 所示。图中 proposed1 是基于扩展窗的逐个选择法, EWF 编码方法的 $\Gamma_1 = 0.084$ 。

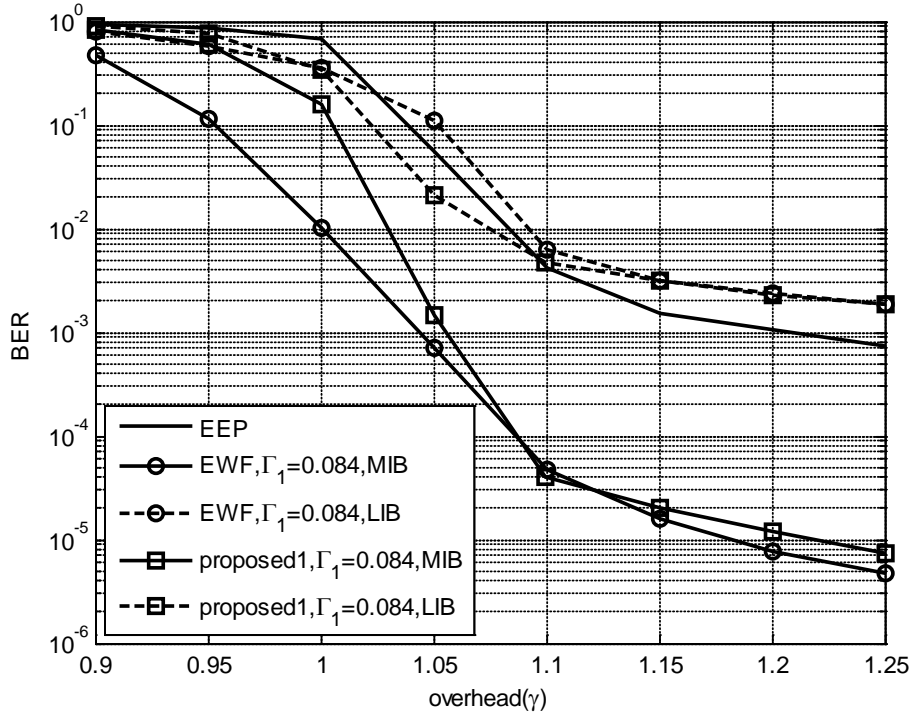


图 3.8 基于扩展窗的逐个选择法与 EWF 法的误码率仿真对比 1

图 3.8 中, 基于扩展窗的逐个选择法的 $\Gamma_1 = 0.084$ 。当译码开销 γ 较小时, 基于扩展窗的逐个选择法的两个等级误码率之间的差异较小, 但是 MIB 集合误码率的下降较快, 当 γ 达到 1.1 左右时, 这两种编码方法的误码性能几乎相同。甚至在 $\gamma = 1.1$ 时, 基于扩展窗的逐个选择法的误码率低于 EWF 编码方法的误码率。

由图 3.8 可知, Γ_1 越大, MIB 集合的误码率越小。图中基于扩展窗的逐个选择法在译码开销 $\gamma = 1.1$ 左右, MIB 集合的误码率与 EWF 编码方法的误码率已经很接近。如果我们增大 Γ_1 , 那么基于扩展窗的逐个选择法的 MIB 集合误码率将有可能低于 EWF 编码方法。

图 3.9 给出了 EWF 编码方法中 $\Gamma_1 = 0.084$, 基于扩展窗的逐个选择法的 $\Gamma_1 = 0.11$ 时, 两者的误码率对比曲线。可以看到, 对于 MIB 集合来说, $\gamma > 1.05$ 时, EWF 的误码率下降趋势变缓, 基于扩展窗的逐个选择法的误码率逐渐优于 EWF。 $\gamma = 1.25$ 时, 基于扩展窗的逐个选择法的 MIB 集合的误码率为 $y_M = 2.157 \times 10^{-6}$, 而使用 EWF 编码方法, MIB 集合的误码率为 $y_M = 4.767 \times 10^{-6}$,

前者优于后者。

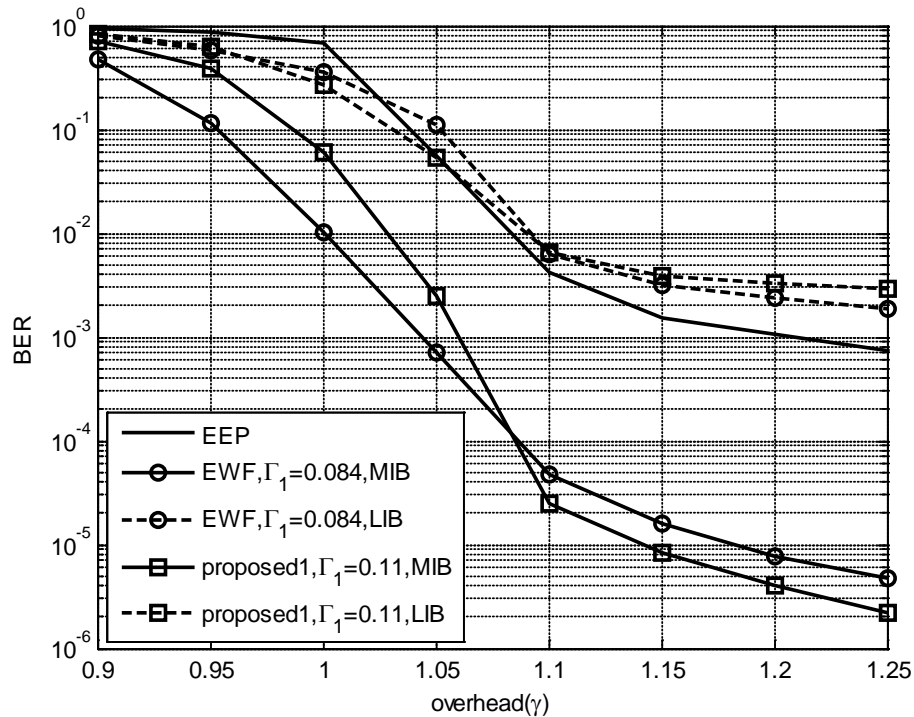


图 3.9 基于扩展窗的逐个选择法与 EWF 法的误码率仿真对比 2

综上，基于扩展窗的逐个选择法虽然在编码复杂度上比 EWF 编码方法大，但是只要选择适当的参数 Γ_1 ，就可以得到比 EWF 编码方法更优的误码性能。

3.4 本章小结

本章提出了一种新的非等差错保护方法，即基于扩展窗的逐个选择法。该方法是用两个窗覆盖全部信息符号，并且都是每选一个符号前都先选择窗，再在窗中选取一个符号，故而称为“逐个选择”。这种先选窗再符号的做法，有效避免了某一个编码符号只包含一个重要等级信息的情况，使得信息的选择更加随机。但是这样也增加了编码的复杂度。

通过理论分析和仿真均证明了基于扩展窗的逐个选择法的参数 Γ_1 越大，MIB 集合的误码率越低，LIB 集合的误码率越高，并且误码率骤降时的译码开销也越大。所以要选择合适的参数 Γ_1 ，不仅要使得 MIB 集合能够达到很低的误码率，又要在较小的译码开销下得到较高的误码性能。基于扩展窗的逐个选择法在仿真中使用的参数 Γ_1 基本满足了上述的要求，并且当 $\Gamma_1 = 0.11$ 时，该方法的误码性能超过了 EWF 编码方法（参数 $\Gamma_1 = 0.084$ ）。

第四章 基于度数的逐个选择法

本章介绍提出的另一种新的非等差错保护方法——基于度数的逐个选择法。该方法从各度值对编码符号的影响出发，试图给出一种误码性能良好并且高优先级信息能够较早恢复出来的编码方法。本章通过逐步试验给出了一种 LT 码的 UEP 编码方法，初步实现了上述目标。

UEP-LT 编码方法和 EWF 编码方法是已知的两种误码性能良好的 LT 码的非等差错保护方法。本章以这两种方法为参照对象，分别与基于度数的逐个选择法的误码性能和数据恢复率进行对比。本章首先介绍 UEP-LT 编码方法，然后通过仿真逐步给出基于度数的逐个选择法，最后对比分析说明本章提出方法的性能。由于 EWF 编码方法已经在第三章介绍，这里不作过多描述。

本章沿用第三章中对各符号意义的定义。

4.1 UEP-LT 编码方法介绍

UEP-LT 编码方法是通过直接增大 MIB 集合中符号的选取概率，使得 LIB 集合的符号选取概率减小，来达到非等差错保护的目的。

UEP-LT 编码方法如下图 4.1 所示。输入的信息符号按照重要等级分为两个集合 S_1 、 S_2 ，每个集合的符号数分别为 $\alpha_1 n$ 、 $\alpha_2 n$ ，其中 $\alpha_2 = 1 - \alpha_1$ 。生成第 i 个编码符号时，按照度分布选取一个度数 d ，再分别选取 d 个输入符号参与编码。选取每个符号前，先以概率 q_1 选取集合 S_1 ，以概率 q_2 选取集合 S_2 ，并且 $q_2 = 1 - q_1$ ，然后在选到的集合中等概随机选取一个符号。按照这样的方法选取 d 个符号，再进行异或，得到一个编码符号 e_i ，($i=1,2,\dots$)。

信息符号：○

编码符号：⊕

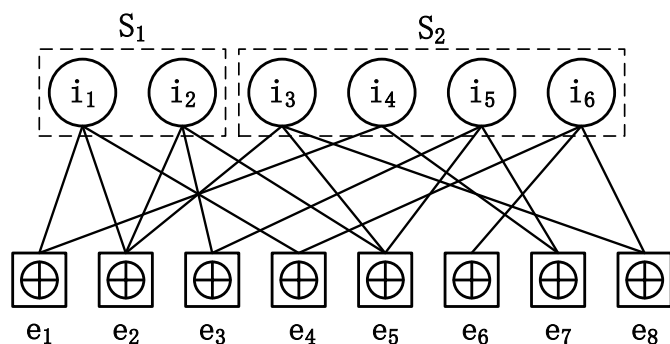


图 4.1 UEP-LT 编码示意图

上述过程中令 $q_i = \alpha_i K_i$, ($i=1,2,\dots,r$), 故有 $\sum_i^r \alpha_i K_i = 1$ 。在这里, 取 $r=2$,

用 K_M 、 K_L 分别对应 K_1 、 K_2 , 所以 $K_L = \frac{1-\alpha_1 K_M}{1-\alpha_1}$ 。 S_1 集合中符号被选中的概率

为 $p_1 = \frac{K_M}{n}$, S_2 集合中符号被选中的概率为 $p_2 = \frac{K_L}{n}$ 。故要实现非等差错保护,

要求 $p_1 > p_2$, 即 $K_M > K_L$ 。

特别的, 当 $K_M = 1$, 即 $K_L = K_M$ 时, UEP-LT 编码就是一般 LT 码的等差错保护编码。

将 q 的值代入到式(2-10), 得到集合 S_j 在第 l 次迭代后的误码率为:

$$\begin{aligned} y_{0,M} &= y_{0,L} = 1 \\ y_{l,M} &= e^{-K_M \cdot \mu \cdot \gamma \cdot \beta (1-(1-\alpha) K_L y_{l-1,L} - \alpha K_M y_{l-1,M})}, l \geq 1 \\ y_{l,L} &= e^{-K_L \cdot \mu \cdot \gamma \cdot \beta (1-(1-\alpha) K_L y_{l-1,L} - \alpha K_M y_{l-1,M})}, l \geq 1 \end{aligned} \quad \text{式(4-1)}$$

由式(4-1)容易得到 UEP-LT 编码的 MIB 集合和 LIB 集合的误码率分别随参数 K_M 及译码开销 γ 变化的理论渐近线, 如图 4.2、图 4.3 所示。

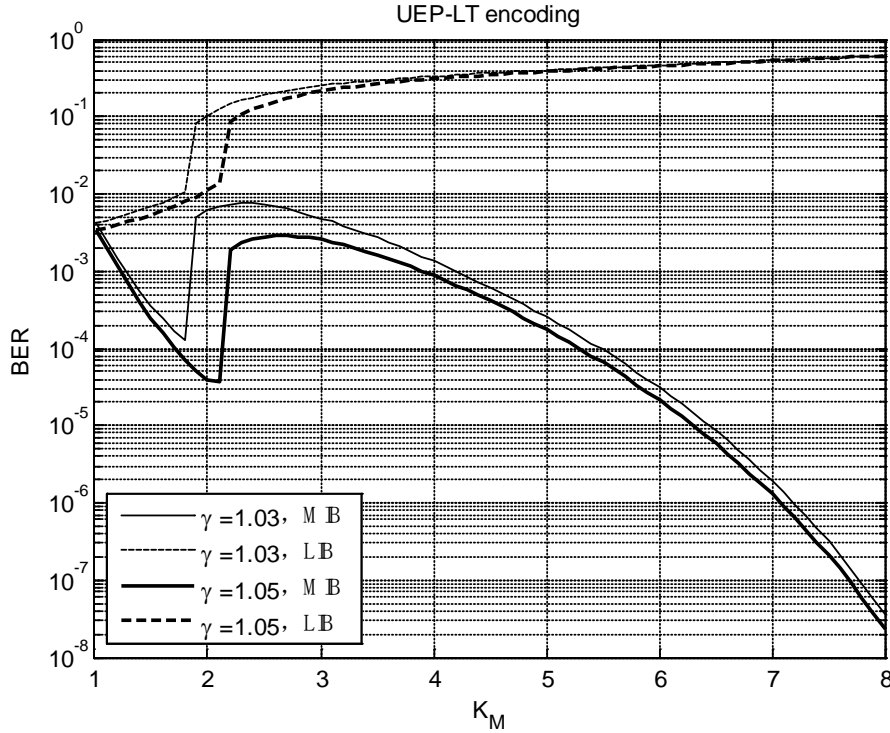


图 4.2 UEP-LT 编码的误码率随 K_M 变化的渐近线

图 4.2 是在译码开销 γ 为 1.05 及 1.03 时 MIB 集合及 LIB 集合的误码性能。

从图中可以看到，无论信息的重要等级如何，译码开销 γ 越大误码率越低。以 $\gamma = 1.05$ 为例，在 $K_M = 2.1$ 处，经过无穷次迭代译码后，MIB 集合的误码率为 $y_{\infty, M} = 3.656 \times 10^{-5}$ ，LIB 集合的误码率为 $y_{\infty, L} = 1.398 \times 10^{-2}$ 。MIB 集合的误码性能高于 EEP 的误码性能，而 LIB 集合的误码性能则低于 EEP。

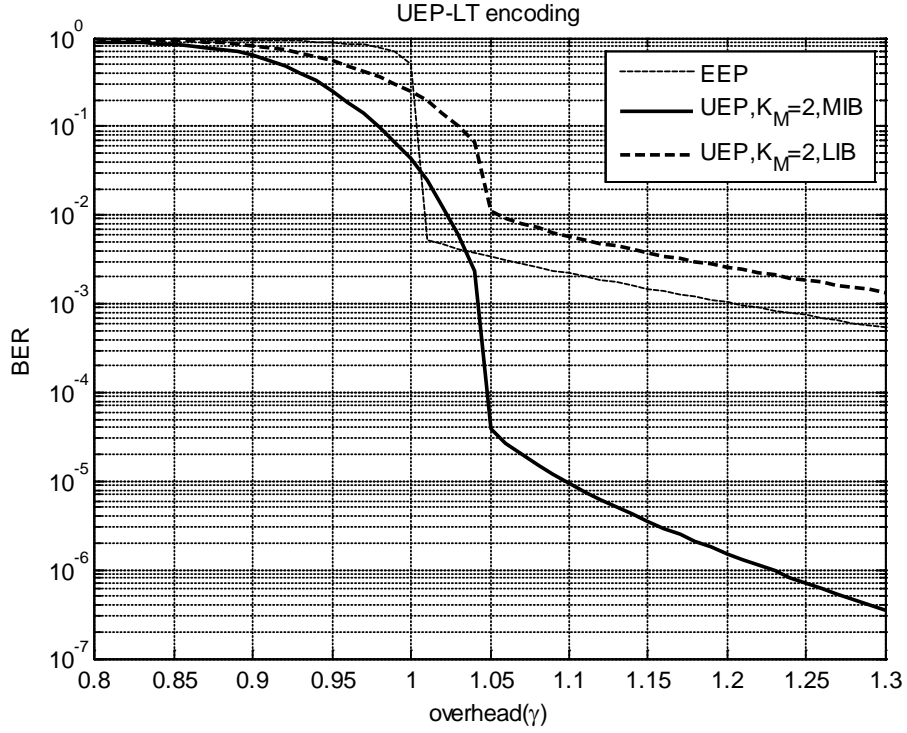


图 4.3 UEP-LT 编码的误码率随译码开销变化的渐近线

图 4.3 显示的是 $K_M = 2$ 时 UEP-LT 编码方法的误码性能。当 γ 较小时，集合 MIB 和 LIB 的误码率下降都很缓慢，两者之间的误码率差距也很小。当 γ 大于 1.05 时，误码率下降的趋势增大，MIB 集合的误码率与 LIB 集合之间的分等级保护越来越明显。当 $\gamma = 1.25$ 时，MIB 集合的误码率极限为 7.147×10^{-7} ，LIB 集合的误码率极限为 1.856×10^{-3} 。相较于 EEP 的 7.436×10^{-4} ，MIB 集合的性能提升，LIB 集合的性能下降，说明采用 UEP 策略，MIB 集合性能的提高是以 LIB 集合性能下降为代价的。

UEP-LT 编码方法的仿真曲线如下图 4.4 所示。仿真时，假设输入 5000 个信息符号，其中有 500 个符号属于 MIB 集合，其它符号属于 LIB 集合。

$$\text{当 } K_M = 2 \text{ 时, } p_1 = \frac{K_M}{n} = 4 \times 10^{-4}, \quad K_L = \frac{1 - \alpha_1 K_M}{1 - \alpha_1} = \frac{8}{9}, \quad p_2 = \frac{K_L}{n} \approx 1.78 \times 10^{-4},$$

故 $p_1 > p_2$ 。 $\gamma < 1.1$ 时误码率下降很迅速，当 $\gamma = 1.1$ 时，MIB 集合的误码率为 $y_M = 1.39 \times 10^{-5}$ ，LIB 集合的误码率为 $y_L = 6.761 \times 10^{-3}$ 。 $\gamma > 1.1$ 时，MIB 集合的误

码率下降趋势变缓, 当译码开销达到 1.25 时, MIB 集合的误码率降到 $y_M = 4.088 \times 10^{-6}$ 。

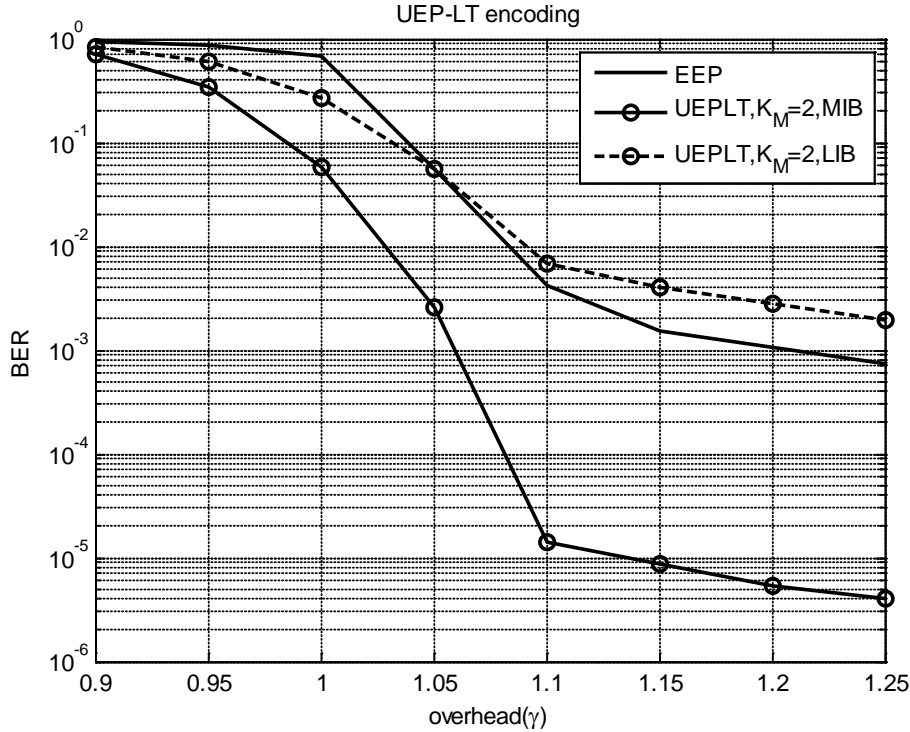


图 4.4 UEP-LT 编码的误码率仿真曲线

4.2 基于度数的逐个选择法

在 LT 码的置信度传播译码算法中, 每次迭代均找度数为 1 的编码符号, 如果没有, 迭代结束, 没有处理过的符号都认为是误码。因此, 在译码时, 一般度数小的符号先处理, 误码也相对小一点。如果为高优先级的符号分配的度数尽量小, 那么高优先级的符号误码率会比较低而且比低优先级的符号先译出来。

但是显然, MIB 集合中的符号要有一定概率参与度数大情况的编码, LIB 集合中的符号也要有一定概率参与度数小情况的编码。

下面我们围绕上述两点, 逐步给出基于度数的逐个选择法。

4.2.1 划分度值法

在设计 UEP 方案时, 首先考虑将随机选择的度值人为地分为小度值和大度值。定义参数 D , 若 $d < D$ 称 d 为小度值; 若 $d \geq D$ 称 d 为大度值。

如图 3.1, 用两个窗覆盖输入信息的全部符号。编码方法如下:

(1) 根据度分布随机选取一个度 d ;

(2) 若 d 是小度值, 以概率 β_1 选择窗 W_1 、以 $1 - \beta_1$ 选择窗 W_2 ; 若 d 是大度值,

以概率 β_2 选择窗 W_1 、以 $1-\beta_2$ 选择窗 W_2 ，在选取的窗中等概随机选择一个符号参与编码；

(3)重复上述步骤，直到选出 d 个符号参与编码，再将它们逐个异或，得到一个编码符号；

(4)依照上面的步骤，不断生成编码符号。

按照上面的编码方法，选择合适的参数 β_1 、 β_2 ，只要使得集合 MIB 大于集合 LIB 的符号选取概率，那么两个集合之间的误码率就会出现差异。设选取小度值的概率为 P_d ， $\alpha_1 = \alpha$ ，易知 MIB 集合的符号选取概率为 $p_1 = \frac{P_d\beta_1 + (1-P_d)\beta_2}{\alpha n}$ ，LIB 集合的符号选取概率为 $p_2 = \frac{P_d(1-\beta_1) + (1-P_d)(1-\beta_2)}{(1-\alpha)n}$ ，令 $p_1 > p_2$ ，可得

$$P_d(\beta_1 - \beta_2) + \beta_2 > \alpha \quad \text{式(4-2)}$$

也就是说选择的参数要满足式(4-2)才能使 MIB 集合的误码率低于 LIB 集合。

取 $D=3$ ，则 $P_d=0.50154$ ，若 $\alpha=0.1$ ，当 $\beta_1=0.15$ ， $\beta_2=0.18$ 时，式(4-2)左侧计算得约 0.165，大于右侧的 0.1；又当 $\beta_1=0.25$ ， $\beta_2=0.1$ 时，计算得 0.175 > 0.1。这两种参数组合都满足式(4-2)，故 MIB 集合的误码率应该低于 LIB 集合。

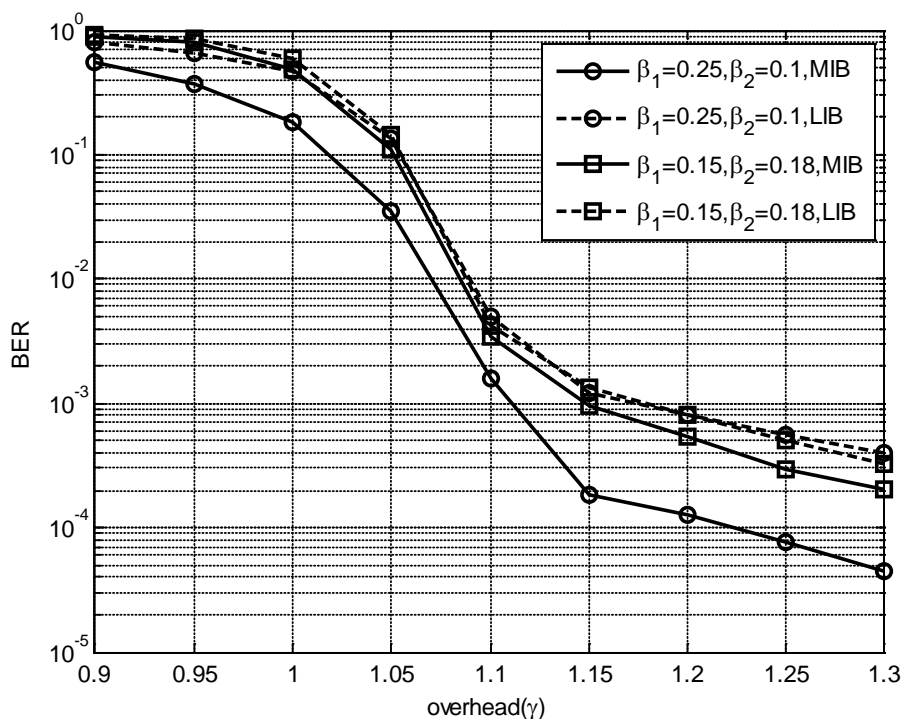


图 4.5 划分度值法的误码率仿真曲线

假设信息符号数为 2000，按照上述的参数组合，仿真得到的误码率曲线如图 4.5 所示。图中， $\beta_1=0.15$ ， $\beta_2=0.18$ 时，MIB 集合与 LIB 集合的误码率几乎没

有差异。 $\beta_1 = 0.25$, $\beta_2 = 0.1$ 时, MIB 集合与 LIB 集合之间的误码率出现差异, 但是这个差异较小, 并且随译码开销的变化不明显。

经过大量的仿真实验, 没有找到合适的 β_1 、 β_2 , 使得集合 MIB 和 LIB 达到更好的 UEP 效果。原因是, 这种人为地将随机度数划分为大小度值的方法缺乏合理性, 没有完全反映度值对编码符号的影响。

因此, 我们进一步考虑将窗 W_1 的选取概率 P_{w_1} 与随机度值 d 关联起来, 即构造一个关于随机度值的函数, 用来表示窗 W_1 的选取概率。于是, 提出了一种新的基于 LT 码的非等差错保护方法——基于度数的逐个选择法。下面将详细介绍这种编码方法及其性能。

4.2.2 基于度数的逐个选择法的编码方法

因为窗 W_1 要在度值小时, 以较大概率被选中, 而在度值大时, 选中概率要较小, 故 P_{w_1} 与 d 成反比。构造 P_{w_1} 与 d 关系如下:

$$P_{w_1} = \frac{1}{\eta(d + f)} \quad \text{式(4-3)}$$

其中 η 为调节系数, f 为偏移量。式(4-3)中, d 越大 P_{w_1} 越小, MIB 集合参与小度数编码的概率比大度数的大, 并且 MIB 集合中的符号也有机会参与较大度数的编码。同时, 只要选择合适的 η 、 f 使得 MIB 集合中符号选取概率大于 LIB 集合的符号选取概率, 就可以实现非等差错保护。

基于度数的逐个选择法编码过程如下:

- (1) 根据度分布随机选取一个度数 d ;
- (2) 由度数 d 计算窗 W_1 的选取概率 P_{w_1} , $P_{w_2} = 1 - P_{w_1}$, 分别以概率 P_{w_1} 、 P_{w_2} 选取窗 W_1 、 W_2 ;
- (3) 在选到的窗中等概地随机选取一个符号;
- (4) 重复(2)、(3), 直到选出 d 个符号, 并将它们逐个异或, 得到一个编码符号;
- (5) 不断重复前面的步骤, 直到编码完成。

假设信息长度为 2000, $\alpha = 0.1$, 对这种编码方法下的 LT 码进行仿真。当 $\eta = 1.4$ 时在 $f = 1$ 及 $f = 2$ 下的仿真结果如图 4.6 所示。图中, 在译码开销 $\gamma = 1.25$ 处, $f = 1$ 时 MIB 集合的误码率达到 9.886×10^{-6} , $f = 2$ 时 MIB 集合的误码率达到 2.237×10^{-5} 。但是译码中 overhead 较高时, MIB 集合会偶尔出现译码完全失效的情况, 限制了 MIB 集合的误码性能。

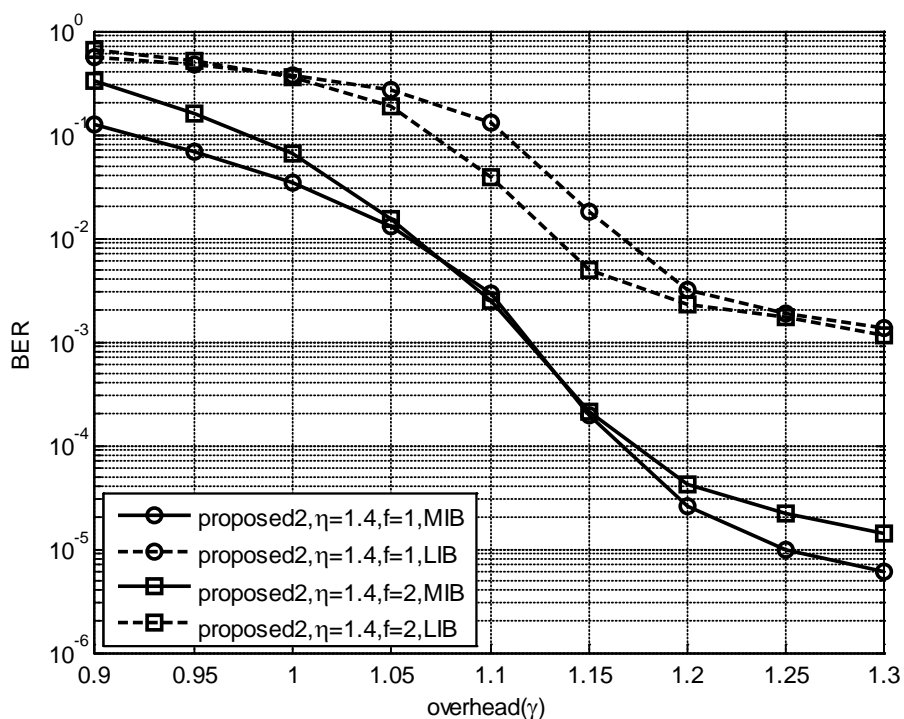


图 4.6 基于度数的逐个选择法的误比特率仿真曲线

基于度数的逐个选择法的初衷是要尽量经过较少的迭代次数恢复出 MIB 集合的符号。图 4.7 显示的是在不同译码开销下，迭代次数与数据恢复率之间的关系。图中的仿真条件为 $\eta=1.4$ ， $f=1$ 。可以看到，要达到相同的数据恢复率，译码开销越大，迭代次数越少。

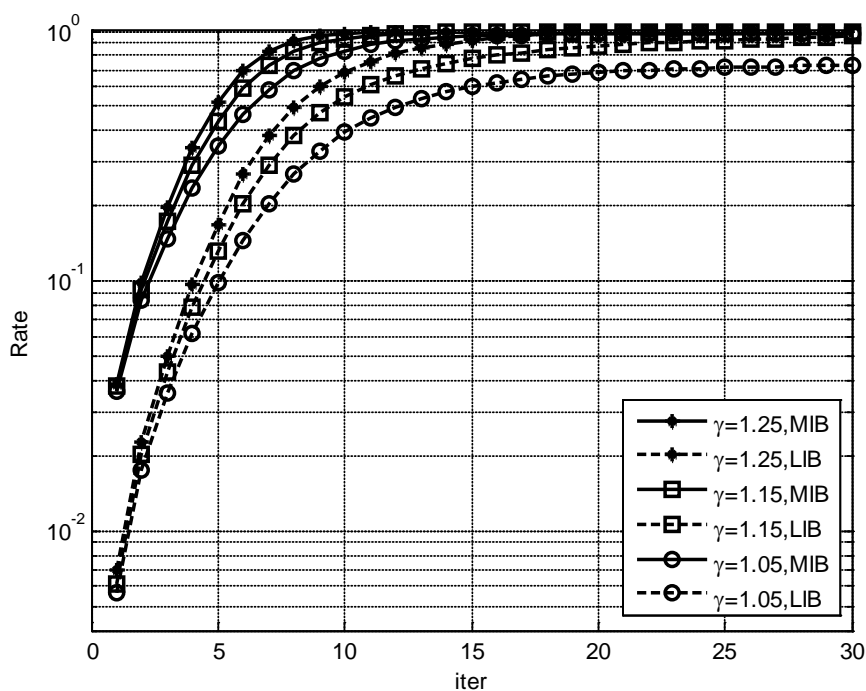


图 4.7 基于度数的逐个选择法的数据恢复率曲线

4.3 基于度数的逐个选择法的仿真对比

本节我们从误码率性能和数据恢复率两个方面，分别说明基于度数的逐个选择法与 UEP-LT 编码方法及 EWF 编码方法相比较的性能。

首先对比基于度数的逐个选择法与 EWF 编码方法及 UEP-LT 编码方法的误码率。仿真中，设输入的信息符号数为 5000，MIB 集合中的符号数为 500。图 4.8 是基于度数的逐个选择法与 EWF 方法的误码率仿真对比曲线，图中 proposed2 就是基于度数的逐个选择法。从图中可以看到，基于度数的逐个选择法在译码开销 γ 较低时误码率比 EWF 编码方法的低，但误码率下降很慢； $\gamma > 1.1$ 时，误码率剧降，当 $\gamma > 1.15$ 时，该方法与 EWF 的误码性能已经非常接近。当 $\gamma = 1.15$ 时，基于度数的逐个选择法的 MIB 集合误码率为 $y_M = 2.115 \times 10^{-5}$ ，而当 $\gamma = 1.25$ 时，该误码率达到 3.982×10^{-6} ，略高于 EWF 方法的 4.767×10^{-6} 。说明，这种方法的误码率性能在译码开销较大时与 EWF 编码方法的误码性能相当。

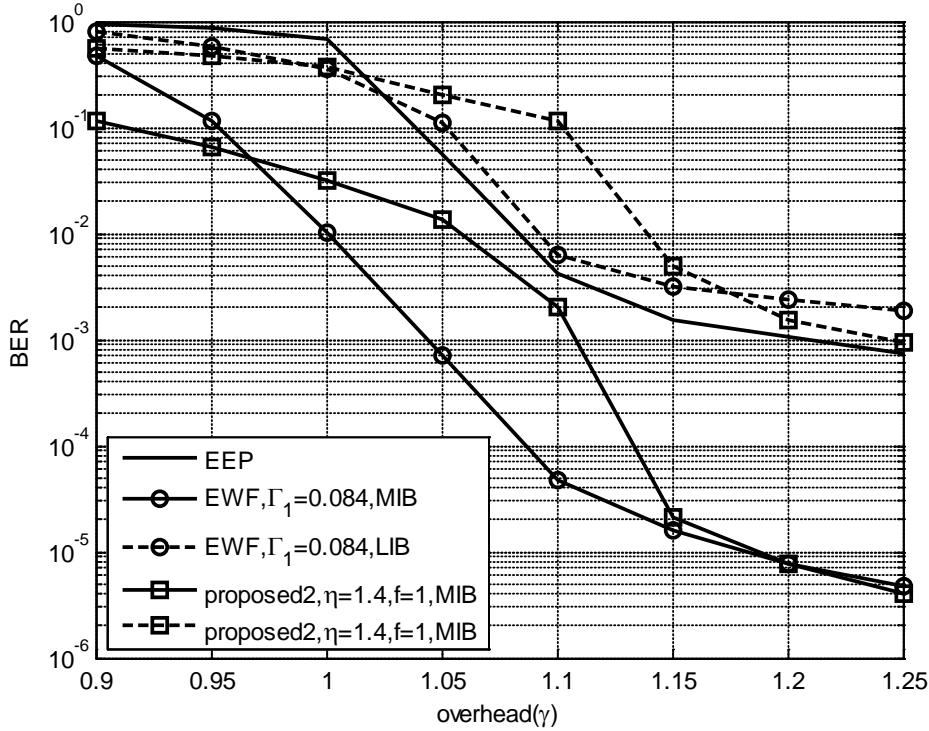


图 4.8 基于度数的逐个选择法与 EWF 编码方法的误码率对比

图 4.9 是基于度数的逐个选择法与 UEP-LT 编码方法的误码率仿真对比。当译码开销 $\gamma < 1$ 时，基于度数的逐个选择法的误码率比 UEP-LT 编码方法的误码率低。但是 $1 < \gamma < 1.05$ 时前者的误码率下降趋势缓慢， $\gamma > 1.05$ 时下降趋势增大，在较高的译码开销处，两者的误码率非常接近。

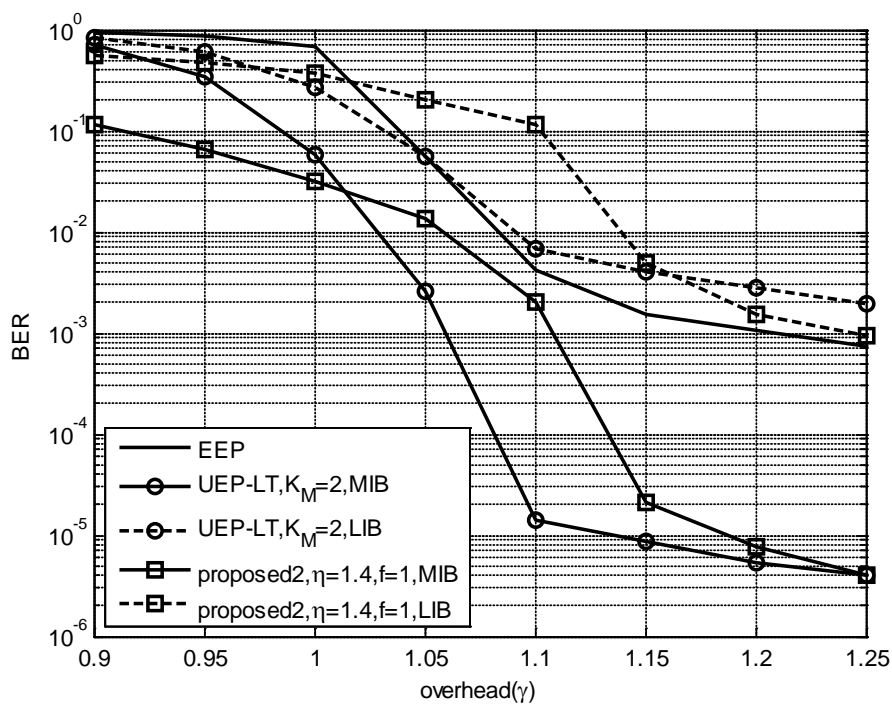


图 4.9 基于度数的逐个选择法与 UEP-LT 编码方法的误码率对比

因为基于度数的逐个选择法的选取原则是 MIB 集合选小度值的概率大，选大度值的概率小，所以这种方法的 MIB 集合符号应该较 LIB 集合更早译出来。

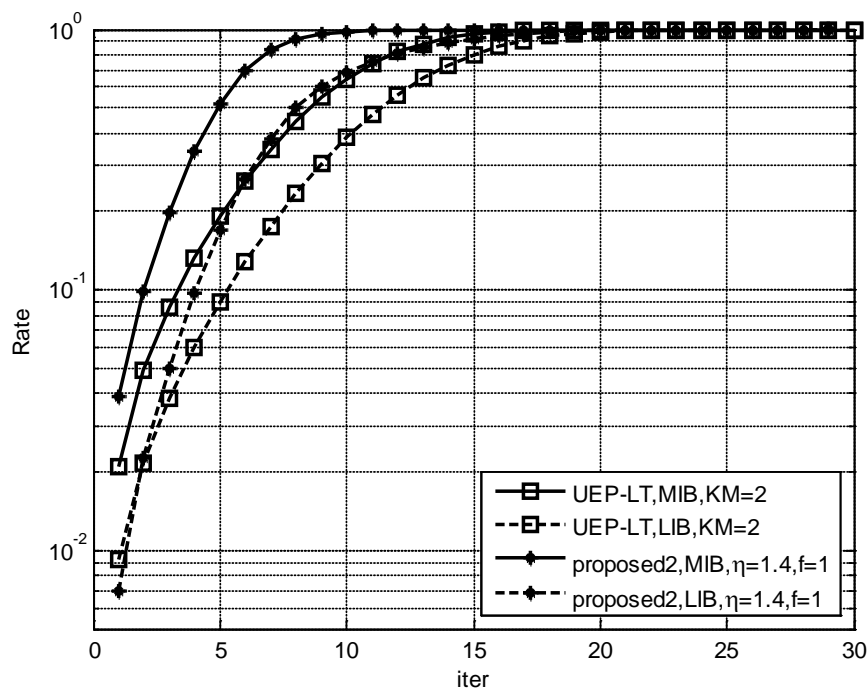


图 4.10 基于度数的逐个选择法与 UEP-LT 编码方法的数据恢复率对比

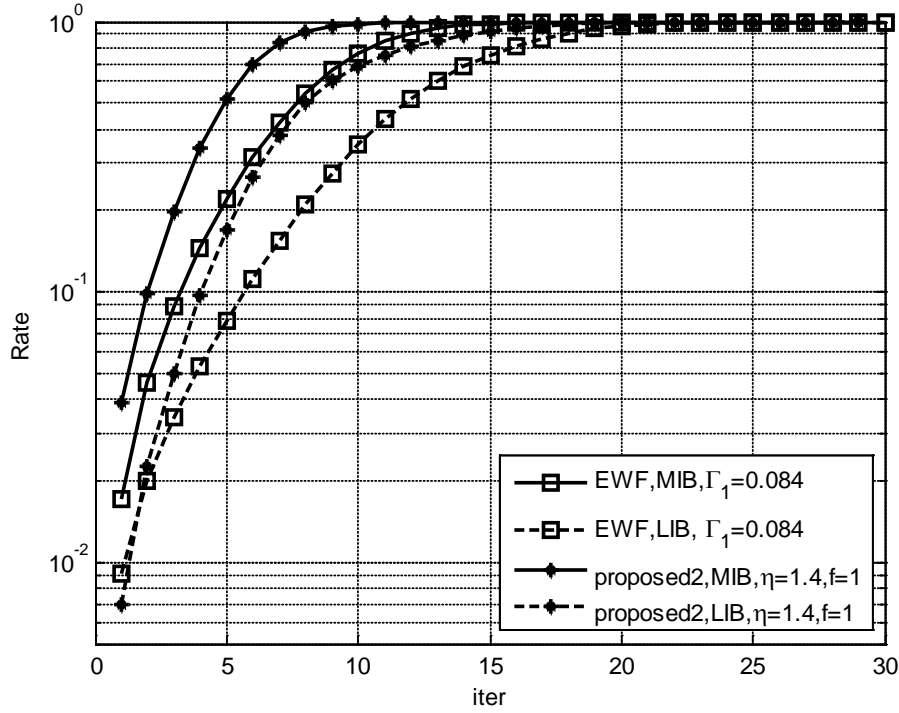


图 4.11 基于度数的逐个选择法与 EWF 编码方法的数据恢复率对比

图 4.10、图 4.11 是基于度数的逐个选择法与 UEP-LT 编码方法及 EWF 编码方法，在译码开销 $\gamma = 1.25$ 时，一次译码中随着迭代次数的增加，集合 MIB 和 LIB 的数据恢复率的对比。图中当 MIB 集合的数据恢复率达到 0.99 时，基于度数的逐个选择法需要迭代 12 次，UEP-LT 编码方法和 EWF 编码方法都需要 20 次。基于度数的逐个选择法的迭代次数减少了 40%。对于 LIB 集合来说，基于度数的逐个选择法的恢复速度也比 UEP-LT 编码方法和 EWF 编码方法的快。

4.4 本章小结

本章提出了第二种基于 LT 码的非等差错保护方法，即基于度数的逐个选择法。这种方法的特点是在较高译码开销下的误码性能与 UEP-LT 编码方法及 EWF 编码方法很接近，并且译码时高优先级的信息能够较快恢复出来。

基于度数的逐个选择法与第三章提出的方法一样，都是用两个窗覆盖全部信息符号，并且都是每选一个符号前都先选择窗，再在窗中选取一个符号。不同的是本章提出的编码方法是根据随机选取的度数 d 确定窗 W_1 的选择概率 P_{w_1} 。仿真发现，基于度数的逐个选择法在译码开销大于 1.15 时，误码性能比 UEP-LT 编码方法和 EWF 编码方法都略差一点；使用基于度数的逐个选择法，达到一定的数

据恢复率, 无论是 MIB 集合还是 LIB 集合需要的迭代次数都比 UEP-LT 编码方法和 EWF 编码方法的少。例如当 $\gamma = 1.25$ 时, 数据恢复率达到 0.99 时该方法比 UEP-LT 编码方法和 EWF 编码方法的译码迭代次数减少了 40%。

第五章 结束语

本文主要对基于 LT 码的非等差错保护方法进行研究。首先介绍了第一类喷泉码——LT 码的编译码原理,并在 BEC 信道下对 LT 码的各项性能进行仿真,包括不同度分布下及不同信息符号长度下的误码性能等,并以重复信息块编码方法为例,介绍了与或树分析法;然后提出了两种新的基于 LT 码的非等差错保护方法,并对它们分别进行了理论分析和仿真对比。

文中在介绍 LT 码的相关知识时,验证了度分布和信息符号长度对 LT 码性能的影响。鲁棒孤波分布下的误码率比弱化的度分布下的低;信息符号越长,误码率下降越快,但是随着译码开销的增大信息符号的长度对误码率的影响逐渐减小。

本文提出的两种新的基于 LT 码的非等差错保护方法,分别在误码性能和数据恢复率方面各有所长。

文中对基于扩展窗的逐个选择法进行了理论分析,给出了误码率渐近线,说明只要选择合适的参数, MIB 集合的误码性能就会优于 EWF 编码方法。并在此基础上,设置不同的参数,对比基于扩展窗的逐个选择法与 EWF 编码方法,验证了在一定的参数下,该方法的 MIB 集合的误码率低于 EWF 编码方法。

提出的第二种 LT 码的 UEP 编码方法,即基于度数的逐个选择法,在较高译码开销下的误码率与 UEP-LT 编码方法及 EWF 编码方法相当,并且通过仿真说明了恢复相同比例信息符号,该方法的迭代次数比 UEP-LT 编码方法和 EWF 编码方法的迭代次数要少。

但是由于时间和精力有限,本文还存在一些问题没有解决,比如:

(1)由于基于度数的逐个选择法中,窗 W_l 的选取概率是根据选择的度值而变化的,如何用与或树分析法对其进行理论分析?

(2)在各种非等差错保护方法下,如何才能选到合适的参数来保证各个集合的误码率性能?选择的依据是什么?

(3)如果增加重要等级的类别,各种非等差错保护方法下的参数要如何选择?特别是基于度数的逐个选择法,要如何构造各个窗的选取概率与度值之间的函数?

致谢

在此硕士论文完成之际，我衷心地向所有在学习上和生活上关心我、帮助我的老师、同学、亲人和朋友表示诚挚的谢意。

衷心感谢我的导师陈彦辉教授。在这两年研究生学习期间，陈老师非常注重对我们自身能力的培养，不但教给了我正确的科研方法和分析问题的思路，还教给了我一些为人处事的道理，无论过去、现在还是将来，我都将受益非浅。陈老师知识渊博，治学态度严谨，思维灵活，而且具有忘我的工作精神，是我今后学习和工作的榜样。

感谢实验室的所有兄弟姐妹们，无论是在科技楼 A614 室，还是在主楼 II 区 340 室，我们都相处融洽，其乐融融。感谢赵恩锋、于文斌、程冰、徐磊、李林、乔百玲、胡伟、钟安宏、李斌、王继荣同学在科研工作和论文研究中给予我的帮助和鼓励。在过去的两年中，大家共同面对求学路上的曲折与艰辛，一起拼搏和奋斗的日子我将终生难忘。

特别感谢父母及家人多年来给我的支持与帮助！他们对我的关心、理解和支持以及生活上的照顾，使我长大成人并顺利完成学业。

谨借此机会向所有给与我关心、支持和帮助的人士表示由衷的感谢！

参考文献

- [1] P. Elias. "Coding for two noisy channels". Proceedings of the 3rd London Symposium on Information Theory, London, U. K., 1955, p. 61-76.
- [2] I. S. Reed, G. Solomon. "Polynomial Codes over Certain Fields". J. Soc. Ind. Appl. Math., p. 300-304, June 1960.
- [3] S. B. Wicker, V. K. Bhargava. "Reed-Solomon Codes and their Applications". IEEE Press, 1994.
- [4] J. Bloemer, M. Kalfane, M. Karpinski, etc. "An XOR-Based Erasure-Resilient Coding Scheme". ICSI TR-95-048, Technical report at ICSI, August 1995.
- [5] M. Sipser, D. Spielman. "Expander Codes". IEEE Transactions on Information Theory. Nov. 1996. Vol. 42, No. 6, pp. 1710-1722.
- [6] J. W. Byers, M. Luby, M. Mitzenmacher, etc. "A Digital Fountain Approach to Reliable Distribution of Bulk Data". Proceedings of ACM SIGCOMM. pp. 56-67, Vancouver, September 1998.
- [7] M. Luby. "LT Codes". Proceedings of the ACM Symposium On Foundations of Computer Science (FOCS). 1, 6-7. 2002.
- [8] A. Shokrollahi. "Raptor codes". Technical report, Digital Fountain. June 2003. DF2003-06-001.
- [9] R. Palanki and J. Yedidia. "Rateless codes on noisy channels". Proc. International Symposium on Information Theory (ISIT) 2004, Chicago, IL, USA, p. 37, June 2004.
- [10] H. Weizheng, L. Huanlin and J. dill. "Digital Fountain Codes System Model and Performance over AWGN and Rayleigh Fading Channels". International Institute of Informatics and Systemics (IIS) on Communications Systems, Technologies and Applications, 2010.
- [11] H. Jenkac, T. Mayer, T. Stockhammer, etc. "Soft decoding of LT-codes for wireless broadcast". Proc. Mobile Summit 2005, Dresden, Germany, June 2005.
- [12] S. Puducheri, J. Kliewer, T.E. Fuja. "Distributed LT Codes". IEEE International Symposium on Information Theory. 2006, pp. 987-991.
- [13] S. Puducheri, J. Kliewer, T.E. Fuja. "The Design and Performance of Distributed LT Codes". IEEE Trans. Inf. Theory. Oct. 2007. Vol. 53, No. 10, pp. 3740-3754.

-
- [14] 朱宏鹏, 张更新, 谢智东. 《喷泉码中 LT 码的次优度分布》. 应用科学学报, 2009, 27(1).
- [15] 龚茂康. 《中短长度 LT 码的展开图构造方法》. 电子与信息学报, 2009, 31(4):885-888.
- [16] Yuan Xiaojui, Li Ping. "Quasi-Systematic Doped LT Codes". IEEE Journal on Selected Areas in Communications. 2009, 27(6): 866-875.
- [17] J. P. Wagner, J. Chakareski and P. Frossard. "Streaming of scalable video from multiple servers using rateless code". Proc. of International Conf. on Multimedia and Expo., 2006.
- [18] N. Rahnavard, B. N. Vellambi, and F. Fekri. "Rateless codes with unequal error protection property". IEEE Trans. Inf. Theory, Vol.53, No. 53, pp. 1521-1532, April 2007.
- [19] D. Sejdinovic, D. Vukobratovic, A. Doufexi, etc. "Expanding window fountain codes for unequal error protection". Proc. 41st Asilomar Conf., Pacific Grove, pp.1020-1024, 2007.
- [20] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi. "Unequal error protection using LT codes and block duplication". Proc. Middle Eastern Multi conference on Simulation and Modeling MESM, 2008.
- [21] M. Nekoui, N. Ranjesh, F. Lahouti. "A Fountain Code Approach towards Priority Encoding Transmission". IEEE Information Theory Workshop. 2006, pp. 52-55.
- [22] D.J.C. MacKay. "Fountain Codes". IEEE Communications. Vol. 152, No. 6, Dec. 2005.
- [23] 黄诚, 易本顺. 《基于抛物线映射的混沌 LT 编码算法》. 电子与信息学报, 2009.10, 31(10). 2527-2531.
- [24] R. Hamzaoui, V. Stankovic, and Z. Xiong. "Optimized error protection of scalable image bitstreams". IEEE Signal Proc. Mag., vol. 22, pp. 91-107, Nov. 2005.
- [25] D. S. Taubman, M. Marcellin. "JPEG2000: Image Compression Fundamentals". Standards and Practice, Norwell, MA: Kluwer, 2001.
- [26] B.-J. Kim, Z. Xiong, and W. A. Pearlman. "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-DSPIHT)". IEEE Trans. Circuits Syst. Video Technol., vol. 10, pp.1365-1374, Dec. 2000.
- [27] M. Luby, M. Mitzenmacher, and A. Shokrollahi. "Analysis of random processes

- via and-or tree evaluation". Proc. 9th SIAM Symp. Discrete Algorithms (SODA), Jan. 1998, pp. 364-373.
- [28] David J.C. MacKay. "Information Theory, Inference, and Learning Algorithms". Cambridge University Press, Version 7.2, March 2005. pp. 589-596.
- [29] (美)S. Lin and D. J. Costello 著. 晏坚, 何元智, 潘亚汉等译. 《差错控制编码》. 北京. 机械工业出版社, 2007. 6, pp. 561-624.



西安电子科技大学

地址：西安市太白南路2号

邮编：710071

网址：www.xidian.edu.cn