

摘 要

低密度校验码以其低复杂度的迭代译码和可逼近香农限而成为目前最佳的编码技术之一，越来越受到众多编码研究者的关注。

本文首先简述了通信系统以及信道编码理论的基础知识，并且对线性分组码的基本原理作了阐述。本文介绍了 LDPC 码的定义、最主要的编译码算法，重点介绍了 DVB-S2 标准中 LDPC 码的特点及编码方法，给出了通过仿真得到的非规则校验矩阵，并对 LDPC 码编码器进行设计并进行仿真。阐述了对数域置信传播算法的几种表达形式，讨论了几种最小和算法的改进算法。然后对修正因子和偏移因子的选择进行了探讨，并仿真证明简化译码算法在运算复杂度大幅下降的情况下性能与标准译码算法相近。

关键词: DVB-S2 标准 LDPC 码 BP 算法 FPGA

ABSTRACT

Low-density parity-check codes come to be one of the best coding technologies because of their low-complexity iterative decoding algorithm and capacity approaching performance ,and they attract more and more researcher's eyes in recent years.

In this thesis, the basis of communication and channel coding theory is introduced briefly. After the definition of LDPC code, the encoding and decoding algorithms, also the performance analysis of decoding algorithms, are studied. Then, it introduces the coding schemes used in DVB-S2 and gets the parity check matrix by simulation. It gives some solutions for designing the encoder. Various log-likelihood-ratio-based Belief-Propagation decoding algorithms and their reduced-complexity derivatives for LDPC codes used in DVB-S2 are presented. Finally, it discusses how to get the correction factor and offset factor using in the Normalized BP-Based decoding and Offset BP-Based decoding. Simulation results show that these reduced-complexity decoding algorithms for LDPC codes achieve a performance very close to that of the BP algorithm.

Keywords: DVB-S2 LDPC BP algorithm FPGA

长春理工大学硕士学位论文原创性声明

本人郑重声明：所呈交的硕士学位论文，《LDPC 码在 DVB-S2 标准中的应用研究》是本人在指导教师的指导下，独立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，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名： 张阳帆 09年3月19日

长春理工大学学位论文版权使用授权书

本学位论文作者及指导教师完全了解“长春理工大学硕士、博士学位论文版权使用规定”，同意长春理工大学保留并向中国科学信息研究所、中国优秀博硕士学位论文全文数据库和 CNKI 系列数据库及其它国家有关部门或机构送交学位论文的复印件和电子版，允许论文被查阅和借阅。本人授权长春理工大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，也可采用影印、缩印或扫描等复制手段保存和汇编学位论文。

作者签名： 张阳帆 09年3月19日

指导导师签名： 陈桂芳 09年3月19日

第一章 绪论

本章介绍了论文的主要研究背景。首先对数字通信的基本模型和信道编码理论发展历程进行了概述，然后介绍了逼近香农限的信道编码技术：LDPC码的提出和研究现状，最后给出全文的内容安排。

1.1 引言

通信系统的目的就是实现发送端信息的无失真传输，在接收端接收到和发送端一样的信息，但是通信信道中普遍存在着信道噪声，尤其是无线通信信道和卫星通信信道，为了使发送端的信息尽可能大的无失真传送到接收端，人们提出了在通信系统的发送端的信息，在发送时，添加一些冗余信息，通过冗余信息来恢复被信道噪声干扰而失真的信息，这就是信道编码和信道译码。但这也造成了信道传输的有效性和可靠性之间的矛盾，而且对于这些冗余信息能在多大程度上恢复失真的信息，和怎样添加冗余信息，都是没有任何回答的。1948年，Shannon^[1]发表了名为“通信的数学理论”具有划时代意义的学术论文。从而从数学的角度解释了怎么添加冗余信息才能最大限度的恢复失真的信源消息，同时也回答了冗余信息能在多大程度上恢复失真的信源信息。他指出了在不可靠的信道下可以进行可靠通信传输的界限和达到这些界限的方法，也就是回答了能在多大程度上恢复信源信息和怎样添加冗余信息的问题。从而彻底改变在之前普遍认为的通信的可靠性和有效性是不可调和的矛盾的观点。

一个数字通信信道可以定义由输入符号 X 和输出符号 Y 以及在信道传输中受到干扰后与之对应的信道传输概率 $p(X/Y)$ 三元关系组成。对于一个有噪声的通信信道，总存在这样一个参数 C ，Shannon证明了当信道传输数据的速率小于 C 的情况下，总存在一种编码技术使得码长充分长时，系统的错误概率可以达到任意小。这就是香农定理。通信系统的系统模型如图1.1

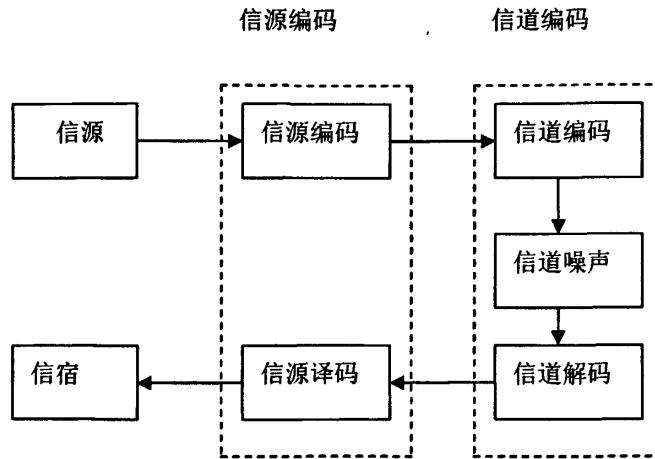


图 1.1 数字通信系统模型

Shannon定理给我们提出了这样的一种保证信息传输不失真的方法即信道编码，信道编码器把将信息源添加一些冗余信息从而形成一个码字，并把这些码字发送出去。这些经过信道编码的信息，受到信道噪声的干扰，必定会产生失真，即从“0”到“1”和从“1”到“0”的变化，在接收端，如果接收到的码字序列，即“0”“1”交替序列和发送端相差不是很大的情况下。接收端就可以根据事先的信道编码规则，利用冗余的信息位，将信道干扰的信号的失真去除掉。从shannon定理我们可以知道，要想达到无误传输的两个条件是：

- (1)构造趋于无穷长的渐进好码或shannon码；
- (2)采用最佳的最大似然译码算法来译码。

几十年来，人们对信道编码做了大量的研究，戈雷码、汉明码、循环码和BCH码等优秀的编码相继被发明。但是他们的性能都无法达到香农码的性能。90年代，信道编码理论取得突飞猛进的发展。93年C.Barrou等人提出的Turbo码^{[7][8]}，这种编码在长码时能够逼近香农限。Turbo码在3G中被采用，取得了巨大的成功。人们由于Turbo码的出现和它取得的巨大成功的影响。对同样采用迭代译码思想译码的早在1962年Gallager发现的低密度校验码(LDPC, Low-Density Parity-Check Codes)^[9]重新重视起来，发现LDPC在长码的条件下也是逼近香农限的。并且在相比较于Turbo码具有三大优点：译码的线性复杂度，抗突变的纠错能力，无交织器带来的延时。

从 Shannon 的噪声信道编码定理非构造性证明中我们可以知道，任何噪声信道都存在好的分组码，实际上所有的分组码都是好的。随着 LDPC 码编解码理论的成熟，越来越多的通信系统都开始采用 LDPC 码作为信道。

1.2 国内外研究现状

1.2.1 LDPC 码的发展历程

早在1963年Gallager在他的博士论文中就提出了LDPC码，并提出了随机构造的方法构造H矩阵，并提出了迭代译码的算法，但由于当时的理论水平和计算机硬件条件的限制，一直没有引起学术界的重视。LDPC码的重新发现和复兴是始于人们对Turbo码迭代算法的深入研究的，同时也是编码理论水平发展和计算机仿真能力的提高的影响。同时，MacKay和Neal，Sipser和Spielman以及Wiberg等人对它重新进行了研究，发现LDPC码同样具有逼近香农限的优异性能。从那时起该码逐渐成为编码领域的一个研究热点。LDPC码是一种特殊的线性分组码，具有一个稀疏的校验矩阵。具有良好的线性复杂度。学者们的研究取得了一定的成果。Zyablov和Pinsker^[16]证明了LDPC码是好码并给出了一种简单的“位翻转”的译码算法。Tanner^[17]提出利用二分图来描述码字的概念，他将LDPC码的校验矩阵与后来称为Tanner图的二分图相对应，在Tanner图上随机构造的LDPC码可以并行进行译码降低了译码的复杂度，同时Tanner还证明了在无环Tanner图上，最小和(Min-Sum)和与积(Sum-Product)译码算法是最优的。

Wiberg在重新研究Tanner的研究成果的基础上，将信息状态引入到Tanner图中，并且与Turbo码和网格图的研究相结合，Tanner的研究成果推广为含有状态变量的Wiberg图，这样便得到了基于网格图的消息传递算法。Mackay和Neal^{[18][11]}发现了“MN”(Mackay and Neal)码，这实际也是一种低密度校验码的重新发现，他们利用Gallager在他的博士论文中提到的随机构造法随机构造出的Tanner图研究LDPC码的性能，采用和积译码算法的规则证明了LDPC码具有与Turbo码具有相似的译码性能，在长码上甚至超过Turbo码。与此同时，Spise和Spieman^[19]研究了扩展码(Expander Graphs)，证明了基于Tanner图的LDPC码为渐进好码。Luby^[20]等人将LDPC码从规则的LDPC码拓展到非规则的LDPC码，并证明了非规则LDPC码比规则LDPC码有更好的性能，当选择好的度序列非规则码的性能甚至要优于Turbo码。Chung等人表明在二进制输入AWGN信道下，设计的码率 $1/2$ 、码长 10^7 的非规则码在误比特率为 10^{-6} 时离Shannon限仅 $0.0045dB$ 。Richardson和Urbanke用称为密度进化理论(Density Evolution Theory)的数值方法来跟踪计算在每次迭代译码中传递的信息的概率密度，并且分析了在码长足够大和足够的迭代次数下，信道参数存在一个阀值现象，当噪声在小于这个阀值下，置信传播算法才能渐进成功译码。他们还进一步综合分析研究了在渐进大的的随机二分图的最佳分布设计问题，并把密度进化理论的分析法延展到更一般的信道中。Kschischang等人在总结Tanner和Wiberg等的研究基础上，建立一个统一的模型(因子图)。在因子图模型中，置信传播及LDPC的迭代译码算法都可归结为基于因子图的和积算法。

1.2.2 LDPC 码的研究现状

自LDPC码自重新受到重视以来，国内外学者对低密度校验码的研究热点主要集中在两个方面，即码本身性能的改进和低密度校验码在各种实际的通信标准中的应用[20][21][22]。

1.2.2.1 LDPC码的构造

低密度校验码之所以能够引起大家的关注，是因为在低密度校验码的校验矩阵是一个低密度的校验矩阵，含有少量的“1”的结构特性决定了低密度校验码的性能，使它具有线性复杂度，并且在由于校验矩阵中只含有少量的“1”因此，相距很远的信息比特参与统一的校验，这就使得低密度校验码自身就具有很好的抗突变性能。并且由于它是一种线性分组码，不需要交织器，避免了编码过程中的延时问题。低密度校验码的一切优异性能都是和它自身的校验矩阵结构相关。因此如何构造H矩阵成为影响低密度校验码性能的关键。目前有关构造LDPC码矩阵H的方法很多，对于长码、中长码、短码具有不同的构造方法，在这些构造方法中，最主要的有两类：Gallager博士在他的博士论文中最终提出的随机构造法和后来人们提出的结构构造法。随机构造法一般适合于长码，其性能能够接近香农限，但是其复杂度过高，在实际应用中限制太大。结构化构造方法大体上可以分为代数构造法和组合构造法。代数方法一般是基于矩阵H的数学变化而来的，它的一般适合用于中，短长度的码。复杂度也较随机构造简单，但是性能稍差。

1.2.2.2 LDPC码的译码

一切编码的好坏都要看解码端，是否能简单准确的译码出发送端送来的信息。这是因为在一般条件下。发送端的条件大多可认为控制，对设备的大型话，设备的大功率化，我们都是可以满足的。而对于特定条件下的接收端，对于译码的复杂度和译码的准确度都要同时满足。低密度校验码的结构(即校验矩阵H)定了之后，译码算法的好坏便决定了编码的性能。译码算法的复杂度决定了工程实现的可能性，而译码算法的准确性决定了工程实施的必要性。目前LDPC码的译码方法主要有两类：一类是基于概率的置信传播，简称BP算法。这类算法的特点是译码性能好，但是译码复杂度高。对接受端硬件要求较高。因此在保留高准确性的同时，研究如何降低译码的复杂度是人们一直研究的问题。因此，研究者提出了许多基于该算法的改进算法。例如对数域BP算法、最小和算法、归一化最小和算法，偏移化最小和算法等。第二类是基于校验和统计迭代的比特翻转译码算法，简称BF算法。这类算法的特点是译码复杂度较低，但同时译码准确度较低，因此保留译码复杂度低，同时提高译码的准确度成为广大学者研究的方向。但这一类译码算法研究较少。

1.2.2.3 LDPC码的性能分析

随着低密度校验码研究的深入，出现了大量的新的理论，包括对低密度校验码性能分析的理论也出现了新的发展。(1)密度进化，对于任何一种通信编码来讲，都有一

个判决门限，当大于某个门限值时，得出正确的译码码字。当然对于一个通信信道编码的统计规律来看，当信噪比大于某个值时，大量的信息比特被错误的译码，最终造成误码率的增大。Gallager博士在他的研究过程中发现，在信道噪声水平低于某个阀值(门限)时，随着码长趋于无穷大时，码的错误概率可以任意逼近零，否则错误概率将大于一个正常数。Richardson等人基于Gallager的思想提出密度进化理论^[23]，提出在和积译码算法的每次迭代信息传递中出现信息错误是低密度校验编码度分布序列和信道参数的函数。(2)高斯近似，密度进化理论的复杂度是相当大的，特别对于信道相当复杂的来说，这就造成了密度进化理论在实际应用中的不可行性。为了在实际应用中，使密度进化理论变得使用，通过比较少的运算就能找到这个阀值(门限)时，提高密度进化算法的计算速度，学者们提出将多维问题转化为高斯密度均值的一维问题，使多维复杂信道变为简单的一维信道，大大简化了计算的复杂度，快速的用计算机线性规划寻找和优化非规则的码。

1.2.2.4 LDPC码的应用

由于低密度校验码的优异性能是最近才受到人们重视和热点研究的。因此在，先期的各种通信标准中，都采用的是同样采用迭代译码算法的Turbo。比如在3G通信标准中。但是LDPC码有其他任何码都无法与相比较的优势:第一，LDPC码具有与码长成线性关系的复杂度。第二，由于LDPC码迭代译码算法为并行算法，便于硬件实现。第三，LDPC码的校验矩阵具有稀疏性，本身即有抗突发差错的特性，不需要引入交织器，避免时延。这些优点使得低密度校验码的应用前景广阔。

1.3 论文的研究的主要内容

第一章为绪论，阐述数字通信系统，回顾了信道编码的发展历程，概述了LDPC码的分析方法和实际应用情况。

第二章在回顾线性分组码的基础上，概述LDPC码的定义及表示方法，详细阐述了LDPC码的通用编译码方法，详细阐述 H 矩阵的构造。

第三章在充分介绍了LDPC码的基础知识上，详细阐述了在DVB-S2标准中采用的LDPC码的特点及编码方法，分析了其达到线性编码复杂度的原因。最后，对符合DVB-S2标准的LDPC码编码器设计进行了初步设计。

第四章详细阐述了LDPC码采用的译码算法及其简化算法的应用研究。

第五章以符合DVB-S2标准，码长为16200比特，码率为1/4的LDPC码为例，对前面各个章节的理论进行仿真验证。

第二章 LDPC 码的编译码原理

LDPC码是一种特殊的线性分组码，可以用生成矩阵和校验矩阵来表征，其校验矩阵具有低密度的特殊性；它又称为稀疏图码，可以用与校验矩阵完全对应的二分(Tanner)图来表征。

2.1 线性分组码概述

按照信息码元和附加的监督码元之间的检验关系可以分为线性和非线性码，信息码元和监督码元之间的关系为线性关系，即监督码元是信息码元的线性组合，则称为线性码。反之，若两者间不存在线性关系，则称为非线性码。我们有限域上讨论线性分组码的构造，具有 q 个元素的有限域用 $GF(q)$ 表示。 $GF(q)$ 上的 n 维线性空间 V_n 中的一个 k 维子空间 $V_{n,k}$ 称为 (n,k) 线性分组码，编码后总码元数目为 n ，信息码元数目为 k ，监督码元数目为 $r = n - k$ 。编码效率 $R = k / n$ ，它是衡量纠错码性能的一个重要指标。一般情况下，监督位越多，检纠错能力越强，但编码效率也随之降低。

对于二进制 (n,k) 线性分组码，信息数量为 2^k 个，可用编码空间的点数为 2^n 个，总共可能的编码方案有 $\binom{2^n}{2^k}$ 种。一个 (n,k) 线性分组码可以看作是由 k 个线性无关的 n 维向量 $\{g_0, g_1, \dots, g_{k-1}\}$ 作为基底的线性组合，写成矩阵形式为：

$$c = m \cdot G \quad (2.1)$$

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & : & p_{0,k} & p_{0,k+1} & \cdots & p_{0,n-1} \\ 0 & 1 & \cdots & 0 & : & p_{1,k} & p_{1,k+1} & \cdots & p_{1,n-1} \\ \vdots & \vdots \\ 0 & 0 & \cdots & 1 & : & p_{k-1,k} & p_{k-1,k+1} & \cdots & p_{k-1,n-1} \end{bmatrix} = \begin{bmatrix} I_{k \times k} & P_{k \times (n-k)} \end{bmatrix} \quad (2.2)$$

m 为包含 k 个信息序列的分组 $\{m_0, m_1, \dots, m_{k-1}\}$ ； c 为编码后的 n 维向量 $\{c_0, c_1, \dots, c_{n-1}\}$ 。 G 称为生成矩阵， I 为 $k \times k$ 维单位矩阵， P 为 $n \times (n-k)$ 维矩阵。这种形式的生成矩阵称为典型生成矩阵。 G 中的每一行都是一个码字，任意 k 个线性无关的码字都可以作为生成矩阵。

将(2.2)式代入(2.1)式可得到：

$$c = m \cdot G = m \cdot [I \ P] = \{m_0, m_1, \dots, m_{k-1}, p_0, p_1, \dots, p_{n-k-1}\} \quad (2.3)$$

由典型生成矩阵得出的码字 c 中，信息位不变，监督位附加于其后，这种码称为系统码。与生成矩阵 G 对应的校验矩阵 H 由 r 行和 n 列组成，每行代表一个线性监督方程的系数。可表示为：

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ h_{r1} & h_{r2} & \cdots & h_{rn} \end{bmatrix} \quad (2.4)$$

与式(2.2)对应的矩阵 H 为: $H = [-P^T \ I_{n-k}]$, $-P^T$ 是 $(n-k) \times k$ 维矩阵, 它是 P 矩阵的转置, “-”表示 $-P^T$ 矩阵中的每一个元素是 P 矩阵中对应元素的逆元, 在二进制情况下, 仍是该元素本身。显然, 由此得到 H 满足:

$$GH^T = [I_k P] \begin{bmatrix} -P \\ I_{n-k} \end{bmatrix} = 0 \quad (2.5)$$

即 G 与 H 的行生成空间互为零空间。

任一矢量 c 是许用码字的充要条件为:

$$Hc^T = 0 \text{ 或 } cH^T = 0 \quad (2.6)$$

某一码字中非零元素的个数称为这个码字的重量, 简称码重。两个码字对应位置上不同元素的个数称为码组的距离, 简称码距, 又称汉明(Hamming)距离。各码组间距离的最小值称为最小码距, 通常用 d_0 表示。 d_0 表明差错控制能力的下限, 与信道编码的检纠错能力密切相关, 有以下三条结论:

- (1) 为检测 e 个误码, 最小码距应满足: $d_0 \geq e+1$;
- (2) 为纠正 t 个误码, 最小码距应满足: $d_0 \geq 2t+1$;
- (3) 为纠正 t 个误码, 同时能检测 e 个误码, 最小码距应满足: $d_0 \geq e+t+1$ ($e > t$)。

如果用 c_{\min} 表示重量最小(d_{\min})的码字, 根据(2.6)式可得到: $c_{\min}H^T = 0$ 。由于线性分组码的最小重量等于最小码距, 故有监督矩阵 H 的 d_{\min} 个列必然是线性相关的, 也就是 H 中线性无关的列最多不超过 $d_{\min}-1$ 个。而 H 的秩至多为 $n-k$, 则必有 $n-k \geq d_{\min}-1$, 因此 d_{\min} 的上界为: $d_{\min} \leq n-k+1$ 。

设发送码字 c 通过有扰信道传输, 将发生错误看成是模2加入某种“错误格式” E 的结果, 则接收码字 $R = c + E$ 。每一码字 c 都必须满足式(2.6)。因此, 对接收到的码字 R 进行检验:

$$RH^T = (c + E)H^T = cH^T + EH^T = EH^T \quad (2.7)$$

式(2.7)说明 RH^T 完全由错误格式 E 决定, 而与发送的码字无关, 它充分反映了信道的干扰情况。

令 $S = RH^T = EH^T$, 称为接收码字 R 的伴随式(或校正子)。线性分组码的伴随式译码可分为以下三步:

- (1) 由接收到的序列 R , 计算伴随式 $S = RH^T$;
- (2) 若 $S = 0$, R 为正确接收码字; 若 S 不为零, 寻找错误格式;

(3) 由错误格式解出码字 $\hat{c} = R - E$ 。

若 (n, k) 线性分组码集合中的 2^k 个码字等概率发出, 设 $[C]$ 为许用码字集合, 当 R 不属于 $[C]$ 时, 就判为错。然后将 R 与 $[C]$ 中所有码字进行汉明距离比较, 选择 $[C]$ 中与 R 汉明距离最小的码字作为发送码字。这种通过选择最小汉明距离来译码的方法, 称为最大似然译码。

2.2 LDPC 码的定义及二分图表示

LDPC码是一种特殊的线性分组码, 因而它可以用校验矩阵来定义。LDPC码的特殊性在于其校验矩阵是“稀疏矩阵”: 它的元素中, 绝大多数元素是0, 只有很少一部分元素是1。根据稀疏矩阵中非零元素在行和列中的数目, 可以将LDPC码分为规则LDPC码和非规则LDPC码; 而根据矩阵中组成元素, 可以将LDPC码分为二元LDPC码和非二元LDPC码, 二元规则LDPC码是指其稀疏校验矩阵的元素在 $GF(2)$ 上取值, 其每个码元参与 j 个校验方程, 而每个校验方程有 k 个码元参与, 这样定义的线性分组码称为二元规则LDPC码, 记为 (N, j, k) 规则LDPC码。图2-1为示例的规则 $(12, 3, 6)$ LDPC码的校验矩阵。

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (2.8)$$

LDPC码除了可以用校验矩阵表示外, 还可以用二部图^[24]表示, 上面的 $(12, 3, 6)$ 规则二部图如图2.1所示:

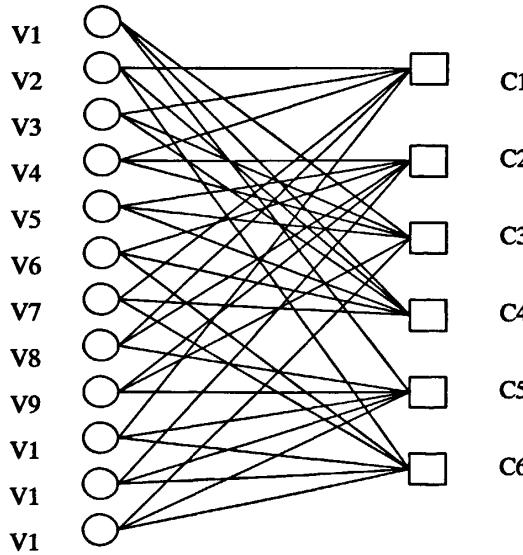


图2.1(12,3,6)规则LDPC码二部图表示

图2.1中的节点分为两类：左边的节点称为变量节点，每个变量节点对应LDPC码的一个码元，其个数等于码长N，也等于校验矩阵的列数；右边的节点称为校验节点，与校验方程对应，其个数等于校验方程的个数，也等于校验矩阵的行数。如果第i个码元参与了第j个校验方程，就用一条边将变量节点 v_i 和校验节点 c_j 相连。与一个节点相连的边数称为这个节点的度。

从二部图的观点来看，一个规则LDPC码相同类型的节点(变量节点或校验节点)具有相同的度，而对非规则LDPC码而言，变量节点(或检验节点)的度不再是相同的，而是服从某个度分布。从校验矩阵的观点来看，一个规则的LDPC码具有相同的行重和相同的列重，而对于非规则LDPC码来说，各行的行重是不相同的，各列的列重也是不相同的。对于非规则LDPC码可以用两类节点的度分布来表示。假设变量节点和校验节点的度分布函数分别为：

$$\lambda(x) = \sum_{i=1}^{d_v} \lambda_i x^{i-1} \quad (2.9)$$

$$\rho(x) = \sum_{j=1}^{d_c} \rho_j x^{j-1} \quad (2.10)$$

其中 λ_i 表示二部图中，从度为i的变量节点出发的边在全部边中所占比例， d_v 表示变量节点的最大度； ρ_j 表示二部图中，从度为j的校验节点出发的边在全部边中所占比例， d_c 表示校验节点的最大度。

如果设 n_i 表示度数为i变量节点的个数， m_j 表示度数为j的校验节点的个数，而E表示二部图中边的总数，那么在给定LDPC码码长N及度分布 $(\lambda(x), \rho(x))$ 的条件下，我们可以计算 n_i ， m_j ， E 如下：

$$n_i = N \frac{\lambda_i / i}{\sum_{i=2}^{d_v} \lambda_i / i} = N \frac{\lambda_i / i}{\int_0^1 \lambda(x) dx} \quad (2.11)$$

$$E = \sum_{i=2}^{d_v} i \cdot n_i = \frac{N}{\int_0^1 \lambda(x) dx} \sum_{i=2}^{d_v} \lambda_i = \frac{N}{\int_0^1 \lambda(x) dx} \quad (2.12)$$

如果设M为校验节点的总数，那么

$$m_i = M \frac{\rho_i / i}{\sum_{i=2}^{d_v} \rho_i / i} = M \frac{\rho_i / i}{\int_0^1 \rho(x) dx} \quad (2.13)$$

同样，我们可以计算得到

$$E = \sum_{i=2}^{d_v} i \cdot m_i = \frac{M}{\int_0^1 \rho(x) dx} \sum_{i=2}^{d_v} \rho_i = \frac{M}{\int_0^1 \rho(x) dx} \quad (2.14)$$

所以有

$$M = N \frac{\int_0^1 \rho(x) dx}{\int_0^1 \lambda(x) dx} \quad (2.15)$$

因而，对于度分布函数为 $(\lambda(x), \rho(x))$ 的LDPC码，其设计码率R为：

$$R = \frac{N - M}{N} = 1 - \frac{\int_0^1 \rho(x) dx}{\int_0^1 \lambda(x) dx} \quad (2.16)$$

从上面的公式可以看出，LDPC码的设计码率是由LDPC码度分布函数决定的。

二元LDPC码是由一个稀疏的0-1矩阵定义的。一个码字是合法的，当且仅当它与校验矩阵的乘积在模2加法和乘法下是一个全0向量。Davey和Mackey将二元LDPC码推广到了一般有限域 $GF(q)$ 上。 $GF(q)$ 上LDPC码同样由一个相似的“稀疏”校验矩阵定义，只是校验矩阵中的元素是从 $GF(q)$ 上取值。一个码字是合法的，当且仅当其与校验矩阵的乘积在 $GF(q)$ 上的加法和乘法运算下为全零向量。

从直观上来看，非规则LDPC码的性能优于规则LDPC码。这是因为在变量节点和校验节点总数一定时，度数大的变量节点从校验节点得到的校验信息较多，因而能够更好的被正确译码，这些译码信息经校验节点提供给度较小的变量节点，使度较小的变量节点也能更好的被译码。理论分析和仿真结果证明了这一点。对 $GF(q)$ 上的LDPC码，已有结果证明其性能较二进制码为优。在LDPC码的直接编码方式中，构造稀疏校验矩阵 H 是首要的问题。下面介绍几种校验矩阵 H 的生成算法。

2.3 校验矩阵 H 的构造

2.3.1 矩阵 H 的半随机(semi-random)构造算法

矩阵 H 的半随机构造算法可以概括为以下步骤:

- (1) 系统首先生成一个 $m \times n$ 维的全0矩阵, 然后随机的往每列插入 j 个1;
- (2) 调整行重, 使行重尽量保持一致;
- (3) 调整列中1的位置, 使得相邻两列1的位置在行上不重叠;
- (4) 消除矩阵中的短循环。

这种算法的运算量比较大, 有时会不收敛, 需要更换随机数种子。构造出的矩阵不一定是规则的, 还需要进一步处理使行重尽可能达到一致。LDPC码在短帧传输情况下, 出现长度为4的环(记为4-环)的概率比较大, 所以必须反复地扫描 H , 直到所有4-环都被消除。并且要避免与变量节点连接的校验节点过于集中。在消除短环过程中, 有两种矩阵 H 的生成方式:

(1) 矩阵 H 的evenboth生成方式

在这种生成方式中, 要严格保持矩阵 H 的列和行的重量分别为 j 和 k 。在消去短环过程中, 移动矩阵 H 中某个1后, 相应地该行或列的重量减一, 而另一行或列的重量增一。再从重量增加的行或列其它位置上的1中选择一个, 调回到原有的行或列中, 以使每行和每列的重量保持一致。然后重新检查此时的矩阵 H 是否出现新环, 若有则重复上述步骤。

(2) 矩阵 H 的evencol生成方式

在这种生成方式中, 只保证列的重量固定不变, 即每个变量节点受到的约束程度是一样的。允许一部分行的重量稍微大些或小些, 平均线重量在 k 附近即可。在消去短环过程中, 系统将某个1在某列内上下移动, 即迫使该变量节点从一个校验方程切换到另一个校验方程。此变量节点参与的校验方程的个数并没有改变, 并没有放松对该码字的约束。然后重新检查此时的矩阵 H 是否出现新环, 若有则重复上述步骤。

在evenboth方式下, 强制每个校验方程监督相同数量的比特; evencol方式则允许一部分校验方程的约束范围稍大, 一部分较小。对于这两种生成方式, 二分图中边的条数是非常接近的。

2.3.2 矩阵 H 的PEG(progressive edge-growth)算法

PEG算法是一种构造二分图的简单有效方法, 它以保持尽可能大的girth为目的, 逐个增加变量节点和校验节点的边。具体操作可描述为:给定变量节点的数目 n 、校验节点的数目 m 和变量节点的分布序列后, 边选择程序开始放置新的边, 选择新边时保证尽可能对girth影响较小; 新边放置好后, 继续搜索下一个边, 直到结束。

PEG算法不但适用于规则LDPC码的构造, 也可应用于非规则LDPC码的构造。它具有较好的实用性, 能构造出girth为8、(1008,3,6)的LDPC码。PEG 算法构造的中、短长度的LDPC码是当前所知具有相同参数的最好的LDPC码之一。

2.3.3 矩阵 H 的比特填充(Bit-filling)算法

比特填充算法能设计出二分图中最小环长较大的 (n, j, k) 形式的LDPC码。在给定校验矩阵 H 的列重 j 、行重 k 、girth为 g 和正整数 n_0 $(n_0 < n)$ 时, 也可设计出校验节点数

目 m 较小, 即码率较大的非规则LDPC码。比特填充算法可描述如下:

- (1) 假设已得到行重均小于等于 k 的矩阵 H , 再增加第 n_0+1 列到矩阵 H 上;
- (2) 设第 n_0+1 列的列重为 j , 并初始为空集合 U_1 , 它是以校验节点为元素的一个集合, 是 $\{1, 2, \dots, m\}$ 的一个子集;
- (3) 进一步假设已经增加了 i 个校验节点到 U_1 集合里 ($1 < i < m$), 在保证约束的情况下再增加第 $i+1$ 个校验节点到 U_1 集合里, 如此循环就可以得到一个完整校验矩阵 H 。

2.3.4 矩阵 H 的扩展比特填充(Extended Bit-filling)算法

扩展比特填充算法能在较低复杂度的情况下构造大girth的校验矩阵, 它是比特填充算法的扩展, 基本上同于比特填充算法, 只在不能增加列时, 可以减小girth到 $g-2$, 之后又可以继续增加下去。

在给定的girth约束条件下, 将元素1逐个放入校验矩阵中, 在整个算法的执行过程中, girth约束一直都是变化的。在算法最初执行时保持girth约束的最大值 g , 直到不降低girth约束就不能在校验矩阵中放1时为止。这时girth约束可以降低(以2递减), 算法在保证新的girth约束的条件下继续, 如此往复循环, 当所有的1都成功放入校验矩阵中或girth约束低于给定的最小值 g 时算法停止。前一种情况表示构造校验矩阵成功, 后一种则表示构造失败。

J.Campello等提出采用扩展比特填充算法来设计具有高码率、高girth的LDPC码。LDPC码结构的设计还有, 基于有限几何学的结构设计, 可将编码简化至循环码, 编码电路可用移位寄存器实现; 基于Ramanujan图和按照Margulis概念构建的规则LDPC码, 其性能优于随机算法构造的LDPC码。随着代数学、几何学的不断向前发展, 利用系统化方法构造LDPC码已受到人们的普遍关注。例如正在发展的极空间里 (α, β) 几何等组合数学新分支就是构造LDPC好码的新方法。

2.4 传统 LDPC 码的编码

由于LDPC码属于线性分组码的一种, 其编码过程可采用线性分组码的通用编码方法, 即由信息序列根据码的生成矩阵来求相应的码字序列。LDPC码是由稀疏的校验矩阵定义的, 如果采用以上编码方法, 需要由校验矩阵 H 得到生成矩阵 G , 这一过程可通过矩阵的行列变换和高斯消去完成。编码由 $c = m \cdot G$ 得到。

2.5 LDPC 码的译码

LDPC码有多种译码方法, 本质上都是基于二部图^[23]的消息迭代译码算法。根据消息迭代过程中传送消息的不同形式, 可以将LDPC码的译码方法分为硬判决译码和软判决译码。如果在译码过程中传送的消息是比特值, 称之为硬判决译码; 如果在译码过

程中传送的消息是与后验概率相关的信息，称之为软判决译码，有时也称为和积译码算法。硬判决译码算法比较简单，但性能稍差；软判决译码计算相对比较复杂，但性能较好。为了平衡性能和计算复杂度，可以将两者结合使用，成为混合译码算法。根据消息迭代过程中传送的消息是否进行了量化及量化所使用的比特数，我们可以将译码方法分为无量化译码和量化译码。硬判决译码可以看成是1比特量化译码。软判决译码可以看成无穷多比特量化译码，而混合译码可以看成变比特量化译码。从量化译码的角度看，硬判决译码和软判决属于同一类译码方法。已有的研究表明，可以用3比特量化取得与和积译码算法非常接近的性能。

假设码字 $c = [c_1, c_2, \dots, c_N]$ 经BPSK调制后在AWGN信道上传输，其噪声服从均值为0、方差为 $N_0/2$ 的高斯分布，且 $y = [y_1, y_2, \dots, y_N]$ 表示相应的接收符号序列。下面我们讨论LDPC码的常用的译码算法。

2.5.1 标准比特反转译码算法

硬判决译码也称为比特反转译码，最初由Gallager根据伴随式译码方案提出的，其实现非常简单。比特反转译码算法的基本思想是利用非法校验的信息迭代更正比特错误。假设 $z = [z_1, z_2, \dots, z_N]$ 表示接收符号序列 $y = [y_1, y_2, \dots, y_N]$ 的硬判决序列，即

$$z_N = \begin{cases} 1 & y_n \leq 0 \\ 0 & y_n \geq 0 \end{cases} \quad (2.17)$$

令 s 表示判决序列 z 的伴随式，则 $s = (s_1, s_2, \dots, s_M) = z \cdot H^T$ 。如果判决序列 z 是一个码字，那么 $s = 0$ ；如果 $s \neq 0$ ，那么说明有错误发生。令 F_n 为比特 n 参与的校验方程的非零伴随式的个数。标准的比特反转算法如下：

- (1) 计算伴随式 $s = (s_1, s_2, \dots, s_M) = z \cdot H^T$
- (2) $n = 1, 2, \dots, N$ 如果 $F_n > \delta$ ，则反转比特 z_n
- (3) 重复步骤1和步骤2直到 $s = 0$ 或达到最大迭代次数。

当参与每一个校验方程的比特数都很少且误码率不是很高时，比特反转译码方法是比较有效的。

2.5.2 软判决译码算法 Belief Propagation (BP)

置信传播算法BP(Belief Propagation Algorithm)是一类较重要的消息传播算法，算法中各个节点之间传递的信息是概率或置信信息，如由变量节点 v_i 传递给校验节点 c_j 的信息是 v_i 取某些值的概率信息，该信息的具体取值依赖于 v_i 的观察值和其他所有与 v_i 相连的校验节点(c_j 除外)在上一轮传递给 v_i 的置信信息，同样，由 c_j 传递给 v_i 的信息也是 c_j 取某些特定值的概率信息，该信息的取值依赖于 c_j 的观察值和其他所有与 c_j 相连的信息节点(v_i 除外)在上一轮传递给 c_j 的置信信息。我们首先介绍两个与BP算法密切相关的定理：

定理1：假设m个独立的二进制数 $\bar{a} = (a_1, a_2, \dots, a_m)$ 中 $p(a_k = 1) = p_k$ 。则序列 \bar{a} 中有偶数个1的概率为： $\frac{1}{2} + \frac{1}{2} \prod_{k=1}^m (1 - 2p_k)$ ，有奇数个1的概率为 $\frac{1}{2} - \frac{1}{2} \prod_{k=1}^m (1 - 2p_k)$ 。

定理2：假设 $y_i = x_i + n_i$ ，其中 n_i 是满足 $(0, \sigma^2)$ 分布的高斯随机变量，并且 $p(x_i = 1) = 0.5$ ， $p(x_i = 0) = 0.5$ ，则 $p(x_i = x | y_i) = \frac{1}{1 + e^{-2xy_i/\sigma^2}}$ ($x \in \{-1, 1\}$)

我们定义编码后产生的码字中第 i 个比特 z_i 的后验概率为：

$$p(z_i = 1 | Y, S_i) \quad (2.18)$$

式中 Y 是接收序列， S_i 表示事件： z_i 所参加的校验式都成立。由条件概率的定义和Bayes公式，我们可以得到 z_i 的后验概率比为：

$$\frac{p(z_i = 0 | Y, S_i)}{p(z_i = 1 | Y, S_i)} = \frac{p(S_i | z_i = 0, Y)}{p(S_i | z_i = 1, Y)} \times \frac{p(z_i = 0, Y)}{p(z_i = 1, Y)} = \frac{p(z_i = 0, Y)}{p(z_i = 1, Y)} \times \frac{1 - p_i}{p_i} \quad (2.19)$$

其中， $p_i = p(z_i = 1 | y_i)$ ； $p_i = p(S_i | z_i = 0, Y)$ 表示的是给定 Y 和 $z_i = 0$ 的情况下， z_i 参加的所有校验式均成立的概率，因此，也就相当于每个校验式中除了 z_i ，其他比特为‘1’的个数为偶数的概率。由定理1可知：

$$p(S_i | z_i = 0, Y) = \prod_{j \in \mathcal{R}(i)} \left\{ \frac{1}{2} + \frac{1}{2} \prod_{i' \in \mathcal{R}(j) \setminus i} (1 - 2p_{i'}) \right\} \quad (2.20)$$

式中 $p_{i'} = p(z_{i'} = 1 | y_{i'})$ ， $z_{i'}$ 是码字中与第 j 个校验式的第 i' 个位置相对应的比特。 $y_{i'}$ 是与 $z_{i'}$ 相对应的接收符号。同理：

$$p(S_i | z_i = 1, Y) = \prod_{j \in \mathcal{R}(i)} \left\{ \frac{1}{2} - \frac{1}{2} \prod_{i' \in \mathcal{R}(j) \setminus i} (1 - 2p_{i'}) \right\} \quad (2.21)$$

将式(2.20)和式(2.21)代入式(2.19)中，我们得到：

$$\frac{p(z_i = 0 | Y, S_i)}{p(z_i = 1 | Y, S_i)} = \frac{1 - p_i}{p_i} \times \frac{\prod_{j \in \mathcal{R}(i)} \left\{ 1 + \prod_{i' \in \mathcal{R}(j) \setminus i} (1 - 2p_{i'}) \right\}}{\prod_{j \in \mathcal{R}(i)} \left\{ 1 - \prod_{i' \in \mathcal{R}(j) \setminus i} (1 - 2p_{i'}) \right\}} \quad (2.22)$$

直接计算式(2.19)比较困难，Gallager给出了一种迭代算法求解式(2.19)。

定义两个变量： $q_{ij}(b)$ 和 $r_{ij}(b)$ 。 $q_{ij}(b)$ 是从变量节点 v_i 传递给校验节点 c_j 的软信息，表示的是给定 y_i ，并且除了第 j 个校验式以外的所有校验式都满足的条件下， $z_i = b$ 的概率。 $r_{ij}(b)$ 是从校验节点 c_j 传递给变量节点 v_i 的软信息，表示的是在 $z_i = b$ 、参加第 j 个校验式的其他比特满足概率 $q_{i'j}(b)$ ($i' \neq i$)的条件下，该校验式成立的概率。通过

上面的定义我们可以很容易的得到 $q_{ij}(b)$ 和 $r_{ij}(b)$ 的计算公式:

$$r_{ij}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i \in \{j\} \cup \{i\}} (1 - 2q_{ij}(1)) \quad (2.23)$$

$$r_{ij}(1) = \frac{1}{2} - \frac{1}{2} \prod_{i \in \{j\} \cup \{i\}} (1 - 2q_{ij}(1)) \quad (2.24)$$

于是式(2.19)改写为:

$$\frac{p(c_i = 0 | \bar{y}, S_i)}{p(c_i = 1 | \bar{y}, S_i)} = \frac{(1 - p_i) \prod_{j \in \{i\}} r_{ji}(0)}{p_i \prod_{j \in \{i\}} r_{ji}(1)} \quad (2.25)$$

$$q_{ij}(0) = K_{ij} (1 - p_i) \prod_{j \in \{i\} \setminus j} r_{ji}(0) \quad (2.26)$$

$$q_{ij}(1) = K_{ij} p_i \prod_{j \in \{i\} \setminus j} r_{ji}(1) \quad (2.27)$$

常数 K_{ij} 是用来保证 $q_{ij}(0) + q_{ij}(1) = 1$ 。根据上面对概率域上的BP算法每一个变量的详细推导, 我们将概率域上BP算法的具体译码步骤归纳如下:

(1) 初始化: 由定理2得到, $q_{ij}^0(0) = p(x_i = 1 | y_i) = \frac{1}{1 + e^{-2y_i/\sigma^2}}$, $q_{ij}^0(1) = 1 - q_{ij}^0(0)$ 初

始化迭代次数 $k = 1$ 。

(2) 根据式(2.23)和式(2.24)计算出从校验节点传递给变量节点的软信息 r_{ji}^k 。

(3) 根据式(2.26)和式(2.27)计算出从变量节点传递给校验节点的软信息 q_{ij}^k

(4) 我们根据式(2.25)计算 z_i 的后验概率: $Q_i^k(0) = K_i (1 - p_i) \prod_{j \in \{i\}} r_{ji}^k(0)$

$Q_i^k(1) = K_i p_i \prod_{j \in \{i\}} r_{ji}^k(1)$, 常数 k_i 保证 $Q_i(0) + Q_i(1) = 1$

5. 对 $Q_i^k(1)$ 进行硬判决, 产生译码结果 $\hat{z}_i = \begin{cases} 1 & Q_i^k(1) > 0.5 \\ 0 & Q_i^k(1) \leq 0.5 \end{cases}$ 将得到的译码

结果序列 \hat{Z}^k 左乘校验矩阵, 获得各个校验式的校验结果: $S^k = (s_1^k, s_2^k, \dots, s_M^k)$ 。

6. 重复2到5, 直到 $S^k = 0$ 或到达最大的迭代次数。

Gallager证明, 如果校验矩阵 H 中没有闭合环路, 当迭代次数趋向无穷大时, $Q_i^k(1)$ 和 $Q_i^k(0)$ 将收敛到 z_i 的先验概率。

第三章 DVB-S2 中 LDPC 码线性时间编码

LDPC码译码性能优良，算法简单。但面临的一个主要问题就是其较高的编码复杂度和编码时延。若按照常规线性分组码的编码方法，通过先得到生成矩阵来编码，涉及到求解奇偶校验方程组的复杂问题，而且对于生成矩阵G并不能保证它是稀疏矩阵，通过这样的方法来编码，算法复杂度是码长的指数关系。这严重制约了LDPC码的应用，然而学者T.J.Richardson和R.L.Urbanke在^[25]文献中指出，对LDPC码的校验矩阵H进行一定的预处理后，LDPC码编码器的计算复杂度与分组码长度成线性关系。实际上，有效的LDPC码编码器的设计是可行的，编码器算法复杂度比译码器算法复杂度还要小。

3.1 LDPC 码的线性时间编码

3.1.1 基于准下三角校验矩阵的LDPC线性编码

在LDPC码的校验矩阵H中，在H中非零元素的个数是码长的几倍到几十倍，它们之间是线性关系。如果能很好的利用校验矩阵的稀疏性，那么运算复杂度只和矩阵H中非零元素的个数有关，就可以实现线性时间编码。根据这种思路，我们采用了Richardson等提出的利用准下三角校验矩阵实现线性时间内编码的方法。

为了保持校验矩阵H的稀疏性，仅对矩阵H执行行列置换，得到图3.1所示

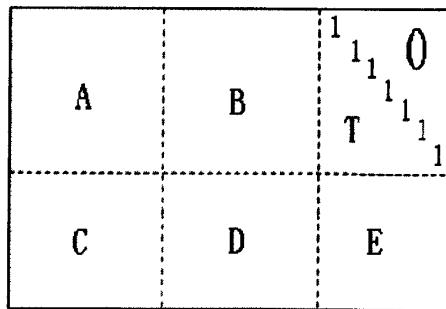


图3.1具有下三角形式的校验矩阵

更确切地将矩阵表示成如下形式：

$$H = \begin{pmatrix} A & B & T \\ C & D & E \end{pmatrix} \quad (3.1)$$

其中A为 $(m-g) \times (n-m)$ 维，B为 $(n-g) \times g$ 维，T为 $(m-g) \times (m-g)$ 维，C为 $g \times (n-m)$ 维，D为 $g \times g$ 维，E为 $g \times (m-g)$ 维，这些矩阵都是稀疏的，T是左上至右下对角线上

为1的下三角矩阵。用 $\begin{pmatrix} I & 0 \\ -ET^{-1} & I \end{pmatrix}$ 从左边乘以这个矩阵，将得到

$$\begin{pmatrix} A & B & T \\ -ET^{-1}A+C & -ET^{-1}B+D & 0 \end{pmatrix} \quad (3.2)$$

定义编码后的码字矢量 $C = (S, p_1, p_2)$, S 为编码的信息比特部分, p_1 (长度为 g) 和 p_2 (长度为 $m-g$) 的组合为编码的校验比特部分, 式(3.1)可分解为以下两个等式:

$$AS^T + Bp_1^T + Tp_2^T = 0 \quad (3.3)$$

$$(-ET^{-1}A+C)S^T + (-ET^{-1}B+D)p_1^T = 0 \quad (3.4)$$

定 义 : $\phi = -ET^{-1}B+D$ (ϕ 为非奇异矩阵), 由式 (3.4) 得出。
 $p_1^T = \phi^{-1}(-ET^{-1}A+C)S^T$ 。一旦 $g \times (n-m)$ 维的矩阵 $-\phi^{-1}(-ET^{-1}A+C)$ 预先计算好之后, 计算 p_1 的复杂度可控为 $O(g * (n-m))$, 当对矩阵执行乘法累加运算时, 复杂度将大大减少。对于 AS^T 来说, 因为 A 是稀疏矩阵, 其运算具有 $O(n)$ 的复杂度。下一步, 用 T^{-1} 乘以这个结果, 由于 $T^{-1}[AS^T] = y^T$ 等价于 $[AS^T] = Ty^T$, 应用后退置换算法时, 运算复杂度可控为 $O(n)$, T 为稀疏下三角阵, 剩余步骤就相当简单了。此时计算 p_1 的复杂度为 $O(n+g^2)$ 。同理, 由式 (3.3) 得到 $p_2^T = -T^{-1}(AS^T + Bp_1^T)$, 计算 p_2 的复杂度也可控为 $O(n)$ 。

从以上的分析可以看出, 采用式(3.1)形式的校验矩阵进行编码时, 具有线性时间复杂度。蚕食算法可以有效地将矩阵 H 化为准下三角形式, 分为以下三个阶段:

(1) 启动: 设 v 为随机选择的度为 2 的变量节点, c_1, c_2 为与其相连的校验节点。首先执行对角线扩展步骤, 这时将与 c_1, c_2 相连的列作为残留矩阵的头两列, 与 c_1, c_2 相连的行作为残留矩阵的头两行。对于给定的初始图, 残留图是消除一些特定边后的结果, 特定边的选择是一些随机实验的结果。然后考虑残留矩阵和相应的残留图, 若残留矩阵包含度为 1 的行, 则接着进行另一对角线扩展步骤。更确切地说, 首先将度为 2 的变量节点通过已知邻接节点转化为度为 1 的变量节点, 然后对于过程中的每一阶段, 选择一个度为 1 的变量节点 (假设至少有一个这样的节点存在) 只对此节点执行对角线扩展步骤;

(2) 主要三角矩阵变化: 考虑启动阶段已成功输出, $(\rho^{res}, \lambda^{res})$ 是长度为 n^{res} 的结果残留图, 可认为是集合 $C^{res}(\lambda^{res}, \rho^{res})$ 的一个元素。考虑另一个集合 $C^n(\lambda, \rho)$ 中的元素, 它与初始图在这些边上是一致的, 假设在这个新构造的图上运行启动阶段并采用同样的随机选择。很容易看到度分布对的顺序和每一步被选择节点的概率都是相同的。所以, 得到的结果残留图将拥有相同的度分布对、长度和生成概率;

(3) 清除操作: 到目前为止距离 g 增大到了 $\frac{\epsilon}{2}\sqrt{n}$, 并且以很高的概率得到了局部下三角矩阵形式, 此时仅剩下很少的校验节点。

对于一个长度为 n 的 $(3,6)$ LDPC 码，其编码复杂度是 n^2 ，然而实际的操作次数不超过 $0.017^2 n^2 + O(n)$ ，由于这个非常小的常系数，即使码长很大，也可产生实际可行的编码器。Gallager 最初定义的许多变体和扩展是可能的，为了构造更接近容量的 LDPC 码，这些扩展也很重要。构造非规则 LDPC 码的线性编码将是下一步的研究重点。

LDPC 码的编码过程尽管有基于准下三角校验矩阵的有效算法，但其复杂度仍然随着 g 的变化而变化。而循环码或准循环码的编码复杂度与码长成线性关系，并且准循环码集合里的码字具有大的最小距离，文献构造了一种基于光正交码的不规则 LDPC(OOC-LDPC) 码。OOC-LDPC 码的二分图中无 4-环，并且是分组长度 m 的准循环码，其系统码的生成矩阵 G 具有如下形式：

$$G = \begin{bmatrix} & (H_m^{-1}H_1)^T \\ I_{n(m-1)} & (H_m^{-1}H_2)^T \\ & \vdots \\ & (H_m^{-1}H_{m-1})^T \end{bmatrix} \quad (3.5)$$

其中是 $I_{n(m-1)}$ 是 $n \times (m-1)$ 阶单位矩阵。

可以看出，OOC-LDPC 编码能使用具有线性复杂度的 $n \times (m-1)$ 级移位寄存器来实现，其编码复杂度与码长成线性关系。

3.1.2 基于校验矩阵重构的 LDPC 编码

文献实现了一种不需要生成矩阵 G 的 LDPC 码编码器设计方法，它充分利用校验矩阵 H 的特点，对其进行分解与重构。将矩阵 H 分解为两个子矩阵， $H = [H^p : H^d]$ 。其中， H^p 是 $(n-k) \times (n-k)$ 维方阵，称为校验位矩阵。 H^d 是 $(n-k) \times k$ 维矩阵，称为信息位矩阵。 H^p 是双对角线形式的上三角子矩阵，具有如下形式：

$$H^p = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

由式(3.6)可以看出，矩阵 H^p 的每列只含一个 1 或者两个 1。 H^d 由 $q \times t$ 个 π -旋转矩阵(记为 π_A)按一定规则构成， π_A 是随机形成的 $m \times m$ 维方阵，由此推知， H^d 为 $qm \times tm$ 维矩阵。矩阵 π_A 的每行和每列均只有一个 1，若加强约束条件，使每条对角线也只有一个 1，这不但不影响矩阵 H^d 的结构，同时也缩小了搜索矩阵 π_A 的范围，能利用现有

的“皇后算法”。将 π_A 顺时针或逆时针旋转 90° 可得到 π_B ，以同样的方式再旋转两个 90°，可依次得到 π_C 和 π_D ，旋转矩阵由此而得名。当分组长度较长时， π_A 的矩阵表示形式是很困难的，可采用 π_A 的矢量表示形式。矢量的每个元素表示矩阵中每列非零元素的位置，计数采用从上到下的顺序。例如，对于 $m=3$ 的 π_A 矩阵，如式(3.7)所示：

$$\pi_A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad (3.7)$$

它所对应的矢量为 $(3, 1, 2)$ ，对 π_A 进行逆时针连续三次的 90° 旋转可分别得到 π_B ， π_C 和 π_D 如下：

$$\pi_B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \pi_C = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \pi_D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3.8)$$

它们所对应的矢量分别为 $(2, 1, 3)$ ， $(2, 3, 1)$ 和 $(1, 3, 2)$ 。利用 π_A 、 π_B 、 π_C 和 π_D 组成码率为 $1/2$ 的矩阵 H^d ，其排列如式(3.9)所示：

$$H^d = \begin{bmatrix} \pi_A & \pi_B & \pi_C & \pi_D \\ \pi_B & \pi_C & \pi_D & \pi_A \\ \pi_C & \pi_D & \pi_A & \pi_B \\ \pi_D & \pi_A & \pi_B & \pi_C \end{bmatrix} \quad (3.9)$$

将矩阵 H^p 和矩阵 H^d 合并，可得到完整的奇偶校验矩阵 H ，如式(3.10)所示：

$$H = [H^p \ H^d] = \begin{bmatrix} 1 & 1 & 0 & 0 & \pi_A & \pi_B & \pi_C & \pi_D \\ 0 & 1 & 1 & 0 & \pi_B & \pi_C & \pi_D & \pi_A \\ & & & \ddots & \ddots & \pi_C & \pi_D & \pi_A & \pi_B \\ & & & 0 & 1 & 1 & & & \\ 0 & & 0 & 0 & 1 & \pi_D & \pi_A & \pi_B & \pi_C \end{bmatrix} \quad (3.10)$$

相应地将码字矢量 C ，分解为校验位矢量 C^p 和信息位矢量 C^d ，即有

$C = [C^p \ C^d]$ ，根据 LDPC 码的编码原理，校验矩阵 H 与码字矢量 C 之间有如下关系：

$$HC^T = [H^p \ H^d] \cdot \begin{bmatrix} C^p \\ C^d \end{bmatrix} = H^p C^p + H^d C^d = 0 \quad (3.11)$$

映射矢量 v 定义为方程式(3.11)的解。因此有：

$$H^p C^p = v = H^d C^d \pmod{2} \quad (3.12)$$

注意 $[H^p]^{-1} = U^p$ ， U^p 是三角矩阵，则：

$$C^p = U^p v \quad (3.13)$$

给定任意信息位矢量 C^d 后, 根据以上阐述可构造出校验位矩阵 H^p 和信息位矩阵 H^d 。利用式(3.12)可计算出映射矢量 v , 再利用式(3.13)就能计算出信息位矢量 C_p 。由于 H^p 是双对角的上三角矩阵, 可对 $H_p C_p$ 采用高斯消去法求得 C^p , 具有线性复杂度。对给定的信息位矢量 $C_d = \{d_i\}$, 根据方程式(3.12)和式(3.13)可求得校验位矢量 $C^p = \{p_i\}$ 。先由 H^p 矩阵的最后一行求出 p_M :

$$p_M = \sum_j h_{M_j}^d d_j \quad (3.14)$$

递推可得:

$$P_{M-i} = P_{M-i+1} + \sum_{j \in (M-i)_i} h^d d_j \quad (3.15)$$

再向上回带可求得 p_i , 从而得到校验位矢量 $C^p = \{p_i\}$ 。

由给定的信息位矢量 C^d 和求得的校验位矢量 C^p , 可得到LDPC编码后的码字矢量 $C = [C^p C^d]$ 。此时, LDPC码的码长为 $(q+t)m$, 信息位长为 tm , 码率为 $t/(q+t)$, 列重量为 t , 行重量为 q 。

3.2 DVB-S2 标准的前向纠错系统

DVB-S 标准是欧洲第一代数字视频广播(DVB)组织制定的卫星数据广播技术规范, 这是一个全球化的卫星传输标准, 目前已被世界绝大多数国家采用。

在过去的十年间, 数字通信领域中的技术革新大量涌现, 尤其是纠错编码技术的研究已取得突破性进展。伴随这一趋势, 同时由于服务商和用户对卫星通信提供更大容量和更多新业务的需求不断增长, DVB 项目组于 2002 年开始着手进行下一代卫星数字视频广播标准的制定, 并于 2003 年公布了针对卫星宽带业务的第二代标准——DVB-S2, 该标准目前已被国际电信联盟和欧洲电信标准协会采纳为正式标准。

DVB-S2 的目标, 是实现最好的信号播送性能以及完整的系统灵活性和合理的接收机复杂性。DVB-S2 标准与 DVB-S 标准相比有以下优势。

与 DVB-S 相比, DVB-S2 标准在带宽利用率方面有质的飞跃, 在相同的功耗水平下增加了 35% 的带宽。这个巨大的进步主要通过三个方面体现出来: 新的纠错编码方式 (LDPC: 低密度奇偶校验码)、新的调制体制 (8PSK、16APSK 和 32APSK) 和新的工作模式 (VCM: 可变编码调制; ACM: 自适应编码调制)。DVB-S2 标准采用 BCH 码与 LDPC 码级联的前向纠错 (FEC) 系统, 其性能接近理论极限。DVB-S2 提供了 $1/4$ 、 $1/3$ 、 $2/5$ 、 $1/2$ 、 $3/5$ 、 $2/3$ 、 $3/4$ 、 $4/5$ 、 $5/6$ 、 $8/9$ 和 $9/10$ 共 11 种纠错编码比率, 以适应不同的调制方式和系统需求。DVB-S2 引入了 64800 和 16200 两种 LDPC 码长, 码长极长是其性能优异 (距香农限仅 0.7dB , 比 DVB-S 标准提高了 3dB)^{[27][28][29][38]} 的原因之一。

前向纠错系统(FEC)是 DVB-S2^{[36][37]}系统中的一个子系统,由外码(BCH)、内码(LDPC)和比特交织(bit interleaving)三部分组成。其输入流是 BBFRAME(基本比特帧),输出流是 FECFRAME(前向纠错帧)。

每个 BBFRAME(K_{bch} 比特)由 FEC 系统处理后产生一个 FECFRAME(n_{ldpc} 比特),外码系统 BCH 码的奇偶校验比特(BCHFEC)加在 BBFRAME 的后面,内码 LDPC 码的奇偶校验比特加在 BCHFEC 的后面,前向纠错系统如图 3.2 所示。表 3-1 和表 3-2 分别给出长帧($n_{ldpc} = 64800$ 比特)和短帧($n_{ldpc} = 16200$ 比特)FEC 编码表参数。

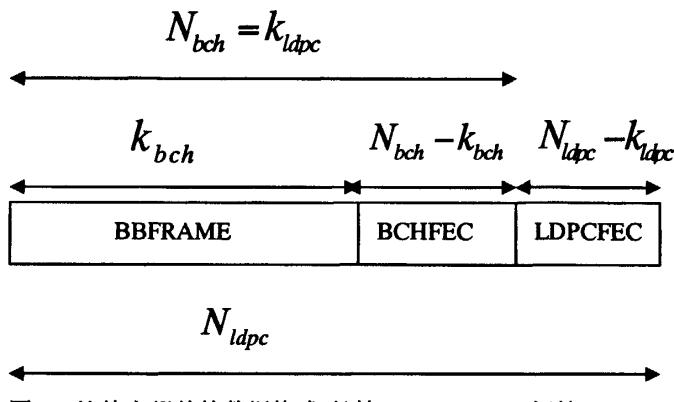


图 3.2 比特交织前的数据格式(长帧 $n_{ldpc} = 64800$, 短帧 $n_{ldpc} = 16200$)

表 3-1 编码参数(一般帧 $n_{ldpc} = 64800$)

LDPC 码率	BCH 信息位 K_{bch}	BCH 码长 N_{bch} LDPC 信息位 k_{ldpc}	BCH 纠错位数	LDPC 码长 n_{ldpc}
1/4	16008	16200	12	64800
1/3	21408	21600	12	64800
2/5	25728	25920	12	64800
1/2	32208	32400	12	64800
3/5	38688	38880	12	64800
2/3	43040	43200	10	64800
3/4	48408	48600	12	64800
4/5	51648	51840	12	64800
5/6	53840	54000	10	64800
8/9	57472	57600	8	64800
9/10	58192	58320	8	64800

表 3-2 编码参数(短帧 $n_{ldpc} = 64800$)

LDPC 码率	BCH 信息位 K_{bch}	BCH 码长 N_{bch} LDPC 信息位 k_{ldpc}	BCH 纠错位数	LDPC 有效码率 $n_{ldpc}/16200$	LDPC 码长 n_{ldpc}
1/4	3072	3240	12	1/5	16200
1/3	5232	5400	12	1/3	16200
2/5	6312	6480	12	2/5	16200
1/2	7032	7200	12	4/9	16200
3/5	9552	9720	12	3/5	16200
2/3	10632	10800	10	2/3	16200
3/4	11712	11880	12	11/15	16200
4/5	12432	12600	12	7/9	16200
5/6	13152	13320	10	37/45	16200
8/9	14232	14400	8	8/9	16200

3.3 DVB-S2 标准中 LDPC 编码算法

尽管 LDPC 码的奇偶校验矩阵是非常稀疏的，在编码过程中，一般的生成矩阵是必须的。当然任何线性分组码都可以通过高斯消去得到生成矩阵，但生成的矩阵会失去稀疏性，而在对于 DVB-S2 标准采用的 LDPC 码，也就是 IRA 码^[30]，由于采用特殊的编码方式，所以不需要存储生成矩阵，具有线性编码复杂度。

根据 DVB-S2 标准，其 LDPC 码的编码任务是由 k_{ldpc} 个信息比特组得到 $n_{ldpc} - k_{ldpc}$ 个奇偶校验比特 $(p_0, p_1 \cdots p_{n_{ldpc} - k_{ldpc} - 1})$ ，最后得到码字 $(i_0, i_1 \cdots i_{k_{ldpc}} - 1, p_0, p_1, \cdots p_{n_{ldpc} - k_{ldpc}})$ 。具体过程概括如下：

第一步，初始化校验位：

$$p_0 = p_1 = \cdots = p_{n-k-1} = 0 \quad (3.16)$$

第二步，计算信息位对奇偶校验位的贡献，计算公式如下：

$$p_j = p_j \oplus i_m, \quad j = \{x + m \bmod 360 \times q\} \bmod (n_{ldpc} - k_{ldpc}) \quad (3.17)$$

其中， p_j 是第 j 个校验位， i_m 是第 m 个信息位， $(n_{ldpc} - k_{ldpc})$ 是奇偶校验位的个数。 x 表示奇偶校验位的地址取 DVB-S2 标准附录 B 和 C 提供的相应地址列表的第 x 行数据。这两个附录分别给出了长码(码长为 64800)的 11 种码率和短码(码长为 16200)的 10 种码率的奇偶校验位地址。 q 是由码率 R 决定的常量。具体方法如下：

(1) 对第一个信息比特 i_0 进行累加, 对应的奇偶节点地址由编码表的第一行指定。DVB-S2 标准一共规定了 21 个编码表, 分别适用于不同的传输要求, 具有不同的码率, 其中长帧 11 个, 短帧 10 个。下面以码率为 2/3 的长帧为例, 说明累加的过程。

$$\begin{aligned}
 P_0 &= P_0 \oplus i_0 & P_{2767} &= P_{2767} \oplus i_0 & P_{10491} &= P_{10491} \oplus i_0 \\
 P_{240} &= P_{240} \oplus i_0 & P_{16043} &= P_{16043} \oplus i_0 & P_{18673} &= P_{18673} \oplus i_0 \\
 P_{506} &= P_{506} \oplus i_0 & P_{9279} &= P_{9279} \oplus i_0 & P_{12826} &= P_{12826} \oplus i_0 \\
 P_{10579} &= P_{10579} \oplus i_0 & P_{8056} &= P_{8056} \oplus i_0 & P_{20928} &= P_{20928} \oplus i_0 \\
 P_{8226} &= P_{8226} \oplus i_0
 \end{aligned} \tag{3.18}$$

(2) 对于下面的 359 个信息比特 i_m , $m = 1, 2, 3 \dots 359$ 累加 i_m 对应的奇偶比特地址为: $\{x + m \bmod 360 \times q\} \bmod (n_{ldpc} - k_{ldpc})$, x 代表和第一个信息比特 i_0 相对应的奇偶比特地址, q 是一个由码率决定的常量, 可作如下简单的推导:

$$q = \frac{N - K}{M} = \frac{N - Nc}{M} = \frac{N}{M}(1 - R) \tag{3.19}$$

对于 DVB-S2 标准 $M = 360$, $N = 64800$ 或 16200。由此得到的 q 值如表 3-3 所示。

表 3-3 一般帧和短帧的 q 值

一般帧 $N = 64800$		短帧 $N = 64800$	
码率	q	码率	q
1/4	135	1/4	36
1/3	120	1/3	30
2/5	108	2/5	27
1/2	90	1/2	25
3/5	72	3/5	18
2/3	60	2/3	15
3/4	45	3/4	12
4/5	36	4/5	10
5/6	30	5/6	8
8/9	20	8/9	5
9/10	18		

继续上面的例子, 码率为 2/3, 故 $q = 60$, 那么对于信息比特 i_1 , 将执行下面的操作:

$$\begin{aligned}
 P_{60} &= P_{60} \oplus i_1 & P_{2827} &= P_{2827} \oplus i_1 & P_{10551} &= P_{10551} \oplus i_1 \\
 P_{300} &= P_{300} \oplus i_1 & P_{16103} &= P_{16103} \oplus i_1 & P_{18733} &= P_{18733} \oplus i_1 \\
 P_{566} &= P_{566} \oplus i_1 & P_{9339} &= P_{9339} \oplus i_1 & P_{12886} &= P_{12886} \oplus i_1 \\
 P_{10639} &= P_{10639} \oplus i_1 & P_{8116} &= P_{8116} \oplus i_1 & P_{20988} &= P_{20988} \oplus i_1 \\
 P_{8286} &= P_{8286} \oplus i_1
 \end{aligned} \tag{3.20}$$

(3) 对于第 361 个信息比特, 被累加的奇偶比特的地址由编码表的第二行决定。用

相同的方式得到接下来的 359 个信息比特所对应的奇偶校验比特的地址。

$$\{x + m \bmod 360 \times q\} \bmod (n_{ldpc} - k_{ldpc}) \quad (3.21)$$

x 代表和信息比特 i_{360} 相对应的奇偶比特的地址,即第二行的第一个值。

(4) 用相似的方法,对每一组 360 个信息比特,利用编码表的一行来找到对应的奇偶比特的地址。

第三步,当所有的信息比特均被使用之后,最终的校验比特节点通过以下方式得到,顺序执行如下操作:

$$p_j = p_j \oplus p_{j-1} \quad (j = 1, 2, \dots, n_{ldpc} - k_{ldpc} - 1) \quad (3.22)$$

最后 p_j 的内容即是奇偶比特 p_j 的值。

这样便得到了码长为 n_{ldpc} 的 LDPC 码码字 $(i_0, i_1 \dots i_{k_{ldpc}} - 1, p_0, p_1, \dots p_{n_{ldpc} - k_{ldpc}})$

图 3.3 给出的是一个符合 DVB-S2 标准的 $1/4$ 码率, 码长为 16200 的 LDPC 码的校验矩阵

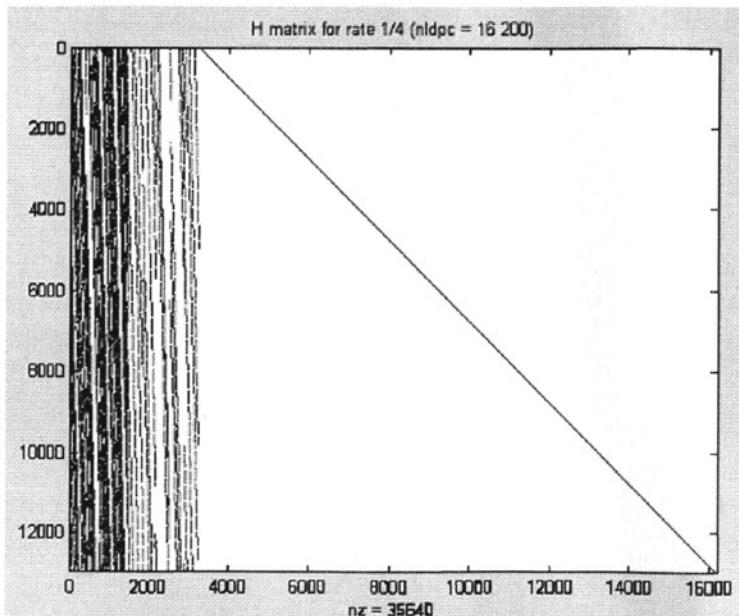


图 3.3 校验矩阵 H

DVB-S2 标准仅仅介绍了编码过程,但是对其采用的 LDPC 码的码型、具体的校验矩阵结构以及其他一些关键参数并不清楚,而这些都与译码能否顺利进行有密切联系,对译码甚为关键。因此我们对 DVB-S2 标准中 LDPC 码的规则进行详细的分析,在此基础上,深入研究 DVB-S2 标准中 LDPC 码的校验矩阵结构,分析得出这是一种 IRA 码。

LDPC 码是线性分组码,当采用信息向量与生成矩阵相乘的方法编码,其编码复杂度与码长平方成正比。码长很大时,编码延时大,限制了 LDPC 码在实时环境下的应

用。因此,如何降低 LDPC 码的编码复杂度一直是该领域工作者努力解决的问题之一。LDPC 码编码方法的研究主要集中在如何利用校验矩阵的稀疏性有效进行编码,以使其编码复杂度随码字长度呈线性增长。

由 DVB-S2 标准可知,校验矩阵 H 包含两部分,具有如下形式

$$H = [H_1, H_2] \quad (3.23)$$

其中 H_1 是一个稀疏的 $m \times k$ 的矩阵,对应着编码规则的第二步,

$$p_j = p_j \oplus i_m, \quad j = \{x + m \bmod 360 \times q\} \bmod (n_{ldpc} - k_{ldpc}) \quad (3.24)$$

即为 H_1 的生成方程。DVB-S2 标准附录 B 和 C 提供的地址列表既给出了 H_1^T 矩阵中 1 的详细位置。编码规则第三步中 $p_j = p_j \oplus p_{j-1}$ 表示的是一个简单的累加器,相应的奇偶校验矩阵每列有两个非零元素,即 H_2 ,它是一个满秩的 $m \times m$ 的矩阵,有如下形式:

$$H_2 = \begin{bmatrix} 1 & & & & & \\ 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & 1 & \\ & & & & 1 & 1 \end{bmatrix} \quad (3.25)$$

以上分析得出 DVB-S2 中的 LDPC 码实际上是 IRA 码。

观察校验矩阵 H 的结构不难看出:代表第一个校验方程的校验矩阵第一行只和第一个校验比特有关,第二个校验方程只和前两个校验比特有关,第三个校验方程只和下两个校验比特有关,依次类推,这样,可以通过一种递推的方式依次得到各个校验比特值。当校验矩阵为以上形式时,生成矩阵可表达为以下系统形式。

$G = [I \ P]$, 其中 $P = H_1^T H_2^{-T}$, H_2^{-T} 具有以下形式

$$H_2^{-T} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ & 1 & \cdots & 1 & 1 \\ & & \ddots & \vdots & \vdots \\ & & & 1 & 1 \\ & & & & 1 \end{bmatrix} \quad (3.26)$$

H_2^{-T} 可以看成另一个编码器,也称为累加器。IRA 码的编码过程可以看成分为两步来完成,首先将码字与一个稀疏矩阵 H_1^T 相乘,得到的值再进行重复累加,得到传输的码字。如图 3.4 所示

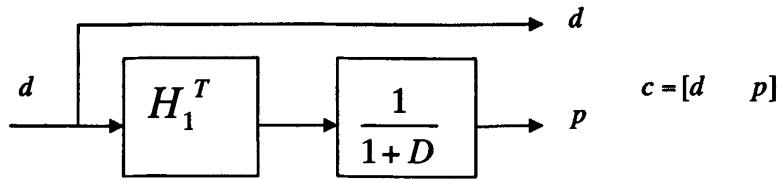


图 3.4 LDPC 码编码器结构

将校验矩阵 H 通过高斯消去得到生成矩阵 $G = [I \ P]$ ，一般 $k \times (n-k)$ 维的矩阵 P 的密度为 0.5，则运算所需的模二加法的数量近似为 $0.5k(n-k)$ ，所以 DVB-S2 中的 LDPC 码编码复杂度较低。

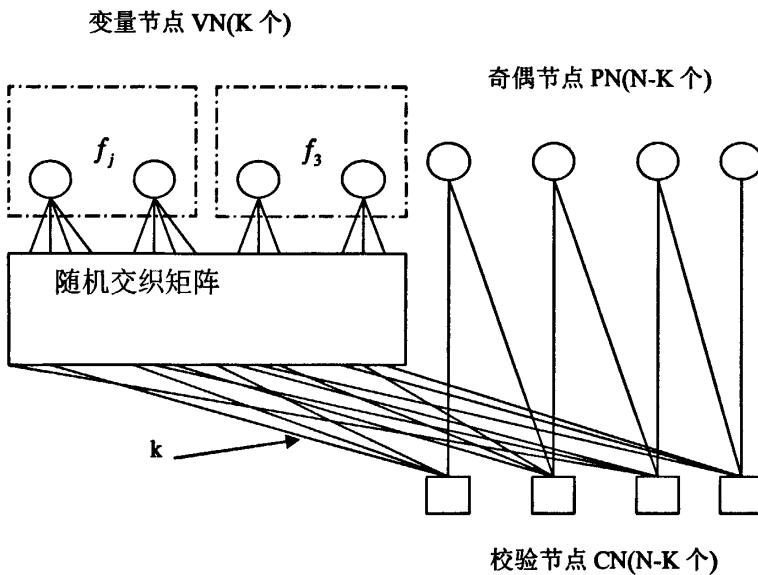


图 3.5 DVB-S2 中 LDPC 码双向图

DVB-S2 中的 LDPC 码对应的双向图如图 3.5 所示。它包含三类节点，包括 K 个信息节点 IN ，对应编码前的信息比特， $N-K$ 个校验节点，对应 $N-K$ 个校验方程， $N-K$ 个奇偶节点，对应 $N-K$ 个奇偶校验比特，信息节点和奇偶节点共同组成 N 个变量节点，对应编码后产生的 N 个比特的系统码字。两类变量节点 VN ： K 个信息节点 IN 和 $N-K$ 个奇偶节点 PN ，分别对应码字中的系统信息位和奇偶校验位。 IN 节点有两种不同的度数 j 和 3，分别用 F_j 和 F_3 表示度数为 j 和 3 的节点数目。校验节点 CN 的数目和与 PN 节点数目相同也为 $N-K$ 个。随机交织矩阵即为 H_1 矩阵，它表示 IN 节点和 CN 节点之间的连接关系，其数学描述为

$$p_j = p_j \oplus i_m, \quad j = \{x + m \bmod 360 \times q\} \bmod (n_{ldpc} - k_{ldpc})$$

若随机交织矩阵的第 i 列和第 j 行有一个非零元素，则 CN 节点 i 和 IN 节点 j 之间有一条边相连。每个 CN 节点恰好和 a 个 IN 节点相连。 PN 节点和 CN 节点之间是一种固定

的“Z”形连接，对应着校验矩阵的 H_2 ，其数学描述为 $p_j = p_j \oplus p_{j-1}$ 。CN 节点的度数为它与 IN 节点和 PN 节点相连的边的总数，则 $N - K$ 个 CN 节点都具有固定的度数 $a + 2$ ，但其值依码长和码率不同而发生改变。表 3-4 和 3-5 分别总结了 DVB-S2 标准中码长为 64800 和 16200 的 LDPC 码在不同码率下的部分参数，表 3-6 总结了 DVB-S2 标准中码长为 64800 和 16200 的 LDPC 码的度分布。这些参数是根据 DVB-S2 标准提供的 q 和标准附录 B 和 C 提供的相应地址列表以及码长和码率得出的。以便有利于我们进一步研究编码器和译码器的结构，以便硬件实现。

表 3-4 DVB-S2 中码长为 64800 的 LDPC 码在不同码率下的 Tanner 图参数

Rate	j	f_j	f_3	$a+2$	m	k
1/4	12	5400	10800	4	48600	16200
1/3	12	7200	14400	5	43200	21600
2/5	12	8640	17280	6	38880	25920
1/2	8	12960	19440	7	32400	32400
3/5	12	12960	25920	11	25920	38880
2/3	13	4320	38880	10	21600	43200
3/4	12	5400	43200	14	16200	48600
4/5	11	6480	45360	18	12960	51840
5/6	13	5400	48600	22	10800	54000
8/9	4	7200	50400	27	7200	57600
9/10	4	6480	51840	30	6480	58320

表 3-5 DVB-S2 中码长为 16200 的 LDPC 码在不同码率下的 Tanner 图参数

Rate	j	f_j	f_3	$a+2$	m	k
1/4	12	1440	1800	4	12960	3240
1/3	12	1800	3600	5	10800	5400
2/5	12	2160	4320	6	9720	6480
1/2	8	1800	5400	6	9000	7200
3/5	12	3240	6480	11	6480	9720
2/3	13	1080	9720	10	5400	10800
3/4	12	360	11520	11	4320	11880
4/5	0	0	12600	13	3600	12600
5/6	13	360	12960	17	2880	13320
8/9	4	1800	12600	27	1800	14400

表 3-6 DVB-S2 中 LDPC 码的度分布

Rate	j	64800			16200		
		λ_j	λ_3	a	λ_j	λ_3	a
1/4	12	2/3	1/3	2	16/21	5/21	2
1/3	12	2/3	1/3	3	2/3	1/3	3
2/5	12	2/3	1/3	4	2/3	1/3	4
1/2	8	16/25	9/25	5	8/17	9/17	4
3/5	12	2/3	1/3	9	2/3	1/3	9
2/3	13	13/40	27/40	8	13/40	27/40	8
3/4	12	1/3	2/3	12	1/9	8/9	9
4/5	11	11/32	21/32	16	0	1	11
5/6	13	13/40	27/40	20	13/121	108/121	16
8/9	4	4/25	21/25	25	4/25	21/25	25
9/10	4	1/7	6/7	28

3.3 编码器设计及实现

LDPC 码的编码通常非常复杂，其复杂度与码长的平方成正比，尤其是对于系统 LDPC 码，因为稀疏的奇偶校验矩阵对应着一个密集的生成矩阵 G 。但 DVB-S2 标准采用的 LDPC 码为 IRA 码，编码复杂度较低，我们采用不通过生成矩阵，直接用校验矩阵来编码，这样的方案的优点是具有线性编码复杂度，而且适合高码率。

3.3.1 编码器设计总体思想

设计编码器时，根据处理中不同的作用可以把他们分成两个功能模块：校验位生成模块和码字输出模块。编码器框图如图 3.6 所示

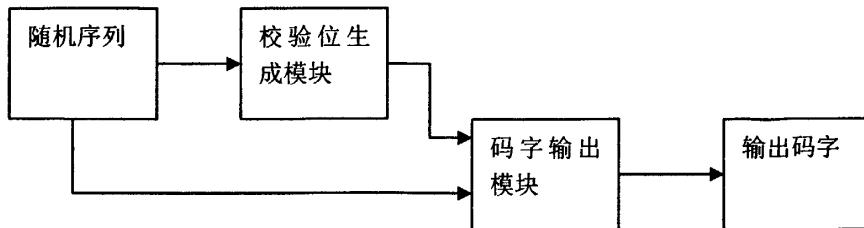


图 3.6 LDPC 码编码器结构

编码器具体算法框图如图 3.7 所示：

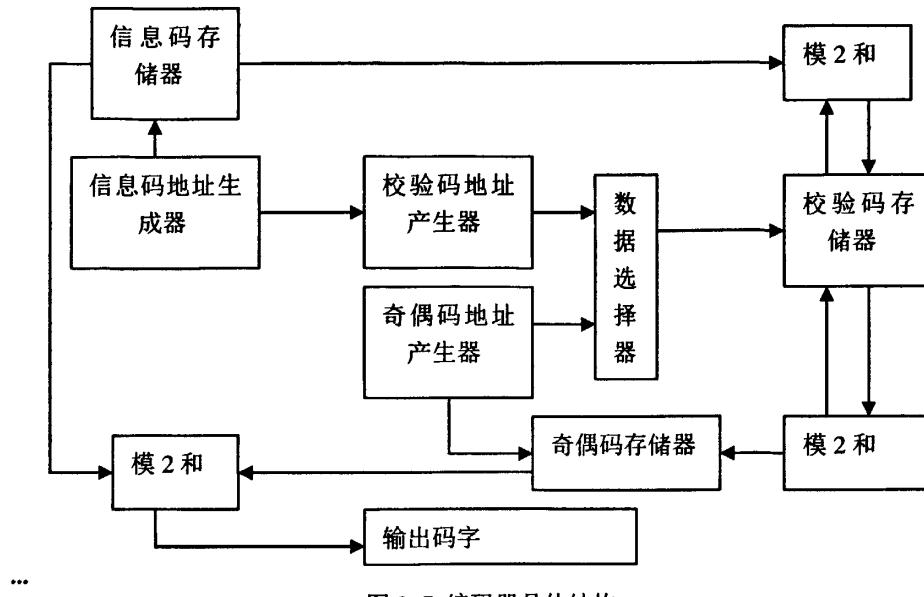


图 3.7 编码器具体结构

编码器的具体编码过程分为两步：

1. 产生校验比特值：根据信息码地址产生器输出的地址 m 从信息码存储器中读取相应的信息比特 i_m ，同时校验码地址产生器根据 m 得到校验比特地址 j ，从校验码存储器中取出相应校验比特 p_j ，和 i_m 进行模二和运算后再存入同一校验比特地址。重复上述过程，直到得到所有校验比特。

2. 产生奇偶比特值：由控制器控制数据选择器和数据分配器转换数据流向。根据奇偶码地址产生器输出的奇偶比特地址 j 从校验码存储器中取出相应校验比特 p_j ，送入累加器中进行重复累加，运算结果再存入同一奇偶比特地址，直到产生所有的奇偶比特，和信息码存储器中的信息比特一起组成系统码字，完成一帧数据的编码。

3.3.2 DVB-S2 中 LDPC 码编译码器的 FPGA 设计^[40]

编码器具体结构^{[31][32]}如图 3.8 所示，校验位生成模块组主要由一个输入 FIFO 处理模块、一个存储计算校验位地址的 RAM 模块和一个存储校验位数值的 RAM 模块构成，主要完成编码器校验位生成、存储和输出的任务。

输入 FIFO 模块的作用是使输入信息比特流无差错地运用于编码处理中，主要由一个异步的 FIFO 构成。由 DVB-S2 中 LDPC 码的编码规则分析可知， H 矩阵中 IN 节点有两种不同的度数 j 和 3 (j 依据 LDPC 码的码率而互不相同)，导致每个信息位参与校验位生成的次数不同。由于比特流输入的连续性，为了防止处理时钟不同而导致输入信息丢失，同时为了保持输入和输出两个时钟域的同步性，需要在输入部分与处理部分之间加入一个异步的时钟 FIFO 进行处理。此模块输出产生的信号 `fifo_one_data` 根据乒乓操作的要求，在不同的时钟处理周期分别传送给不同的存储校验位数值的 RAM 模

块。

同时向存储校验位数值的 RAM 输入的是由计算校验位地址的 RAM 模块输出的地址值。设计中用一个单独的只读 RAM 存储 DVB-S2 标准附录 B 和 C 提供的相应码率的地址列表的行数据。根据编码规则(3.17)，每次需要信息位的贡献地址时，取出对应地址的行数据用于校验位的地址计算即可，达到了校验矩阵有效存储的目的。

存储校验位数值的 RAM 模块由两块双口 RAM 构成，主要用于校验位生成的计算。因为在 DVB-S2 中校验位的生成分为两步。

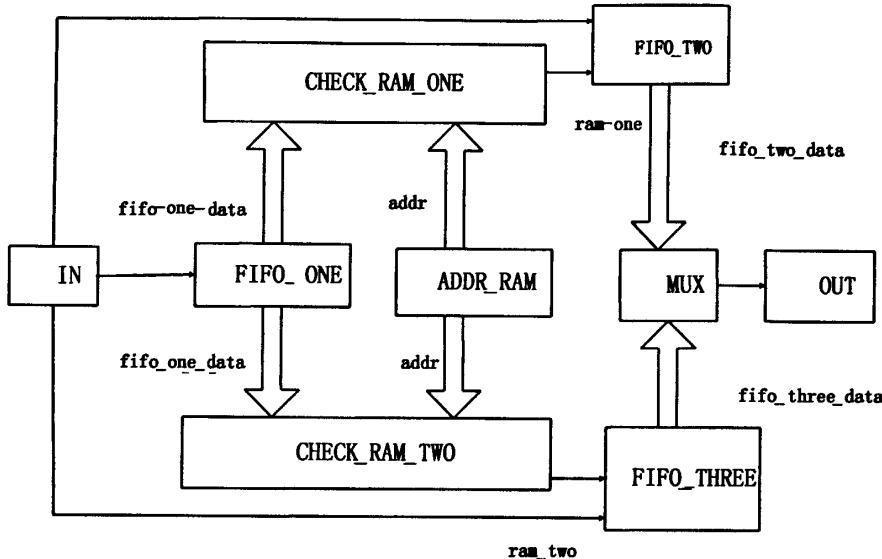


图 3.8 基于 FPGA 的编码器原理图

首先是通过相应信息位的模 2 累加（工作原理如图 3.8 所示），当校验 RAM 第一个时钟接收到地址 $addr$ ，将读出口对应地址的数值 ram_data 读出，送入寄存器 REG；第二时刻接收到的 $fifo_one_data$ 与寄存器的数值相加从写入口写入到第一时刻 $addr$ 对应的地址，即完成一次信息位的累加。

第二步主要是实现编码规则中的 $p_i = p_i \oplus p_{i-1}$ 。原理和信息位的累加一样，主要是运用双口 RAM 的两个端口同时读取前后两个地址的校验位值，然后用模 2 累加器累加生成。

码字输出模块设计是根据乒乓操作的思路，运用两块异步的 FIFO 和一个二选一的 MUX(选择器)配合使用，目的是将流入编码器的信息位和编码生成器生成的校验位连续输出，同时保证处理时钟域和输出时钟域的同步性。

我们采用的通用器件是 altera 公司 Cyclone ii 的 FPGA 芯片，并在 ISE 下使用 VHDL 语言编写源程序，最后得出综合结果。

3.3.3 仿真结果分析

图 3.9 和图 3.10 给出了用 FPGA 实现的该编码器的 Modelsim 仿真波形。图中 clk 是控制输入和编码器数据处理的时钟，每个输入数据对应 4 个 clk 时钟周期。 in 是输入的待编码数据。 clk_two 是控制数据输出的时钟，每个输出数据对应一个 clk_two 时钟

周期。out 是输出的编码后的数据。为了适应数据处理的要求, clk 与 clk_two 的周期为 3: 8。rst 为系统复位使能信号。该编码器实现的是 DVB-S2 中码长为 16200 的 IRA 码编码功能, 码长较长, 由于数据处理的需要, 要在输入 16200 个数据后才会有编码结果输出。图 3.9 是编码器刚开始工作的一段数据, 此时还没有编码结果, 故 out 信号持续低电平。图 3.10 是开始有编码输出时的一段数据。由于 DVB-S2 中的 LDPC 码为系统码, 编码结果由信息位和奇偶校验位组成, 信息位即为输入的待编码数据。比较两图, 可以看出, 图 3.9 中的输入数据 in 与图 3.10 中的输出数据 out 完全一致, 而且通过程序验证, 编码后的奇偶校验位也是正确的, 编码器工作正常。

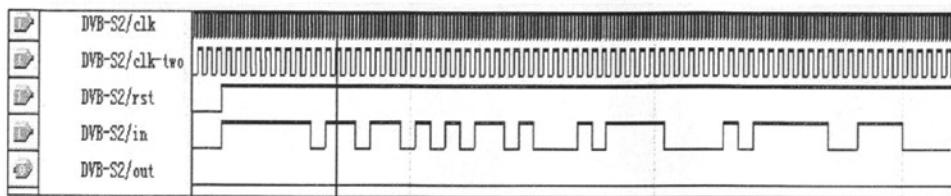


图 3.9 起始仿真波形

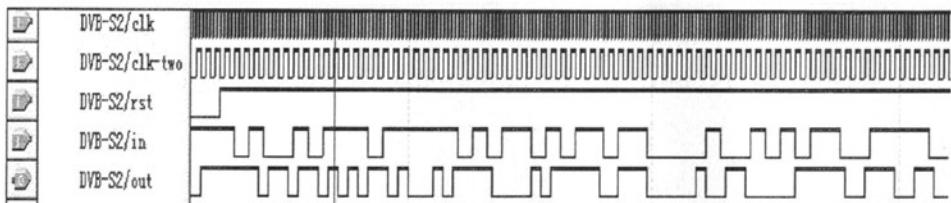


图 3.10 编码后仿真波形

第四章 LDPC 改进译码算法研究

4.1 硬判决译码算法改进

4.1.1 加权比特反转(WBF)译码算法

Kou Yu等人提出了一种适合各种信道的加权比特反转(WBF)算法，在每轮迭中，对每个变量节点按某种方法计算其可靠性，可靠性最小的节点发生反转。WBF算法在性能指标上优于标准比特反转译码算法，但引入可靠性计算导致译码复杂度增加。WBF算法如下：

(1) 对 $m = 1, 2, \dots, M$, 令 $|y|_{\min-m} = \min_{n \in N(m)} |y_n|$

(2) 计算伴随式 $s = (s_1, s_2, \dots, s_M) = z \cdot H^T$

(3) 对 $n = 1, 2, \dots, N$, 计算 $E_n = \sum_{m \in M(n)} (2s_m - 1)|y|_{\min-m}$

(4) 如果 $n = \arg \max_{1 \leq n \leq N} E_n$ 则反转比特 z_n

(5) 重复上述步骤知道 $s = 0$ 或达到最大迭代次数。

由于WBF算法在计算变量节点的可靠性时只考虑了校验节点的信息，ZhangJuntan等人提出了改进的加权比特反转(IWBF)算法，在计算变量节点可靠性时加入了变量节点的信息，提高了译码性能。Guo Feng等人注意到WBF和IWBF算法中只考虑了校验节点的特定信息，其提出的RR-WBF算法考虑了检验节点的全部信息，进一步提高了性能，而没有增加复杂度。

4.1.2 Shuffled 比特反转(SBF)译码算法

标准BF算法通常需要很多次的迭代译码过程才能收敛。为了加快译码器的收敛，可以考虑对编码比特串行处理，即在每个时间单元上只处理一个比特。假设在迭代译码过程的第 n 个时间里第 n 个比特被处理，如果比特的反转条件满足的话，那么这个比特被反转。通常， z_n 反转后新得到的值比其旧值正确的概率要大，从而基于 z_n 反转后的新值计算得到的伴随式 s 比基于 z_n 的旧值计算得到的伴随式 s 要可靠些。在SBF迭代译码过程中，一旦一个比特发生反转，那么这个比特参与的所有校验对应的伴随式的值也将反转，其余比特依据新得到的伴随式进行处理。因为在SBF算法中，可靠的比特值被尽可能早的利用，因而SBF算法比标准的BF算法具有更快的收敛速度。SBF算法描述如下：

(1) 计算伴随式 $s = (s_1, s_2, \dots, s_M) = z \cdot H^T$

(2) 对 $n = 1, 2, \dots, N$, 如果 $F_n > \delta$, 则反转比特 z_n , 且反转 $\{s_m : m \in M(n)\}$

(3) 重复步骤1和2直到 $s = 0$ 或达到最大迭代次数。

从译码性能的角度来说,一个完全串行实现的SBF算法是我们期待的,但是在硬件实现中却是不现实的,特别是当码字的长度很大时更是如此。为了便于硬件实现,可以将比特进行分组,组与组间串行处理,此译码方案称为GSBF方案。在GSBF方案中,比特分组顺序依次处理。显然,在一次迭代完成后,最后处理的那组比特是最可靠的,因为他们利用了前面所有被更新过了的比特信息。

在上述的GSBF方案中,对同一种比特分组方案,如果采用两个译码器对比特分组方案按不同的顺序进行处理,那么将得到同一比特分组方案的不同的可靠性排序。为了利用这一特点,文献提出了一种RGSBF方案,即使用两个或多个译码器,对同一比特分组方案按不同的分组次序进行译码,并在迭代译码过程中互相交换比特译码信息。RGSBF译码快但代价是使用了多个译码器。

4.1.3 其他比特反转译码算法

在LDPC码的硬判决译码方法中,除了上述几类方法外,还有一些基于其他思想的译码算法。为了避免二部图中环形问题而引入的基于Taboo的译码算法,为解决Taboo译码算法中的Taboo参数问题引入的基于令牌的译码算法等。在这些算法中,所有问题的本质在于如何找到合适的反转比特的顺序以使LDPC码的性能达到最优以及降低译码复杂度。

4.2 软判决译码算法改进

4.2.1 对数域上BP算法

在概率域上的BP算法^[33]中,计算 $p_i(d)$ 以及更新 r_{ji}^t 、 q_{ij}^t 时,需要大量的乘法、除法及对数运算,这不仅增加了复杂度,而且会使得算法的数值稳定性变差,限制BP算法的实际应用。如同BCJR算法一样,我们考虑在对数域上实现BP算法,连乘的运算被转换为复杂度更低、稳定性更好的加法运算。在对数域的BP算法中,我们定义以下变量代替式(2.23),式(2.24),式(2.26),式(2.27)中的各个软信息:

$$\begin{aligned} L(r_{ji}) &= \log\left(\frac{r_{ji}(0)}{r_{ji}(1)}\right) & L(q_{ji}) &= \log\left(\frac{q_{ji}(0)}{q_{ji}(1)}\right) & L(Q_i) &= \log\left(\frac{Q_i(0)}{Q_i(1)}\right) \\ L(p_i) &= \log\left(\frac{p(x_i=1|y_i)}{p(x_i=-1|y_i)}\right) = \log\left(\frac{(1+e^{2y_i/\sigma^2})}{(1+e^{-2y_i/\sigma^2})}\right) = 2y_i/\sigma^2 \end{aligned} \quad (4.1)$$

下面我们对对数域的BP算法进行详细的推导。

$$\text{因为 } \tanh(x) = \frac{\exp(2x)-1}{\exp(2x)+1} \quad x = \frac{1}{2} \ln\left(\frac{p_0}{p_1}\right), \quad p_0 + p_1 = 1$$

$$\tanh\left(\frac{1}{2} \ln\left(\frac{p_0}{p_1}\right)\right) = \frac{\sqrt{p_0/p_1} - \sqrt{p_1/p_0}}{\sqrt{p_0/p_1} + \sqrt{p_1/p_0}} = \frac{p_0 - p_1}{p_0 + p_1} \quad (4.2)$$

再根据式(3.7)和式(3.8), 我们可以得到 $L(r_{ji})$ 和 $L(q_{ji})$ 的关系式:

$$\tanh\left(\frac{1}{2}L(r_{ji})\right) = \prod_{i \in R(j) \setminus i} (q_{ri}(0) - q_{ri}(1)) = \prod_{i \in R(j) \setminus i} \left(\frac{1}{2}L(q_{ri})\right) \quad (4.3)$$

$$L(r_{ji}) = 2 \tanh^{-1} \left(\prod_{i \in R(j) \setminus i} \tanh\left(\frac{1}{2}L(q_{ri})\right) \right) \quad (4.4)$$

式 (4.4) 中仍然存在着大量的乘法运算, 为了使得计算更加简单, 我们定义 $L(q_{ri}) = \alpha_{ri} \cdot \beta_{ri}$, 其中 $\alpha_{ri} = \text{sign}(L(q_{ri}))$ $\beta_{ri} = |L(q_{ri})|$ 将式(4.4)改写为:

$$\begin{aligned} L(r_{ji}) &= \left(\prod_{i \in R(j) \setminus i} \alpha_{ri} \right) \cdot 2 \tanh^{-1} \log^{-1} \sum_{i \in R(j) \setminus i} \log \left(\tanh\left(\frac{1}{2}\beta_{ri}\right) \right) \\ &= \left(\prod_{i \in R(j) \setminus i} \alpha_{ri} \right) \cdot \phi \left(\sum_{i \in R(j) \setminus i} \phi(\beta_{ri}) \right) \end{aligned} \quad (4.5)$$

$$\phi(x) = -\log \left(\tanh\left(\frac{1}{2}x\right) \right) = \log \frac{e^x + 1}{e^x - 1} \quad \text{式(3.16)中需式(3.15)中的连乘被加法代替。}$$

在对数域的BP算法中, $L(q_{ij})$ 的计算公式变为:

$$L(q_{ij}) = L(p_i) + \sum_{j \in R(i) \setminus j} L(r_{ji}) \quad (4.6)$$

综上所述, 我们得到对数域上的 BP 算法的具体步骤:

(1) 初始化: $L(q_{ij}^0) = L(p_i) = 2y_i / \sigma^2$, 迭代次数 $k=1$ 。

(2) 由式(3.16) (4.5)计算得到校验节点传递给变量节点的软信息 $L(r_{ji}^k)$ 。

(3) 由式(4.6)计算得到变量节点传递给校验节点的软信息 $L(q_{ij}^k)$ 。

(4) 计算 z_i 的后验概率 $L(Q_i^k) = L(p_i) + \sum_{j \in R(i)} L(r_{ji}^k)$

(5) 对 $L(Q_i^k)$ 进行硬判, 产生译码结果

$$\hat{z}_i = \begin{cases} 1 & L(Q_i^k) < 0 \\ 0 & L(Q_i^k) \geq 0 \end{cases} \quad \text{将得到的译码结果序列 } \hat{Z}^k \text{ 左乘校验矩阵, 获得各个校验式的校验结果: } S^k = (s_1^k, s_2^k, \dots, s_M^k)。$$

6. 重复2到5, 直到 $S^k = 0$ 或到达最大的迭代次数。

无论是概率域的BP算法还是对数域的BP算法, 它们所需的存储量基本相等, 因为校验矩阵的稀疏性, 两者在实现时所要的存储量与码长成线性关系。但是对数域上的BP算法避免了大量的乘法运算, 大大减少了运算复杂度, 更有利于硬件实现。

4.2.2 最小和(Min-Sum)译码算法

Min-Sum 算法^{[34][35]}和 BP 算法比较较相似, 在对数域 BP 算法中, 定义了

$\phi(x) = \log \frac{e^x + 1}{e^x - 1}$ 。该函数满足 $\phi^{-1}(x) = \phi(x)$ ，因此(4.5)中的 $\phi\left(\sum_{i \in R(j) \setminus i} \phi(\beta_{i,j})\right)$ 通过某种近似简化计算。图4.1是函数 $\phi(x)$ 的曲线，从中可以看出当 $|x|$ 趋向于0的充分小的区间里，

$\phi(x)$ 上升的非常快，所以 $\sum_{i \in R(j) \setminus i} \phi(\beta_{i,j})$ 几乎完全由 $\beta_{i,j}$ 中数值最小的一个值决定。

$$\phi\left(\sum_{i \in R(j) \setminus i} \phi(\beta_{i,j})\right) \approx \phi(\min \phi(\beta_{i,j})) \quad (4.7)$$

$$L(r_{ji}) = \left(\prod_{i \in R(j) \setminus i} \alpha_{i,j} \right) \cdot \phi\left(\sum_{i \in R(j) \setminus i} \phi(\beta_{i,j})\right) = \left(\prod_{i \in R(j) \setminus i} \alpha_{i,j} \right) \cdot \phi(\min \phi(\beta_{i,j})) \quad (4.8)$$

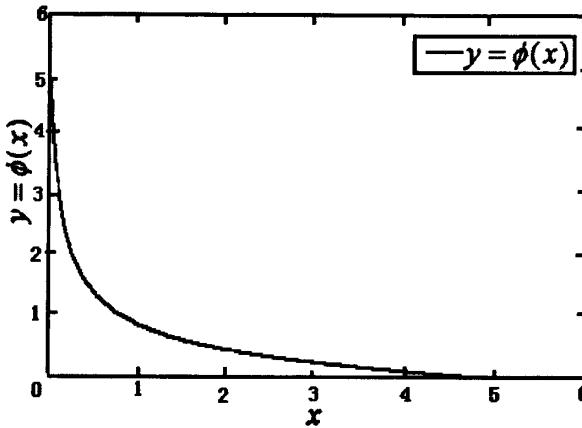


图4.1 函数 $\phi(x) = \log \frac{e^x + 1}{e^x - 1}$

在计算 $L(q_{ij}^0)$ $L(p_i^0)$ 的时候，因为后面的计算只需要对这些数据做求最小值的计算，所以，所以对于高斯白噪声信道的方差 σ^2 也可以省略掉。

$$L(p_i^0) = L(q_{ij}^0) = y_i \quad (4.9)$$

另外 $L(r_{ji}^k)$ 和 $L(q_{ij}^k)$ 这两个变量的更新算法，和对数域BP算法基本相同。

综上所述，我们得到Min-Sum算法的具体步骤：

(1) 初始化: $L(q_{ij}^0) = L(p_i) = y_i$ ，迭代次数 $k=1$ 。

(2) 由式 (4.8)计算得到校验节点传递给变量节点的软信息 $L(r_{ji}^k)$ 。

(3) 由式 (4.6)计算得到变量节点传递给校验节点的软信息 $L(q_{ij}^k)$ 。

(4) 计算 z_i 的后验概率 $L(Q_i^k) = L(p_i) + \sum_{j \in R(i)} L(r_{ji}^k)$

(5) 对 $L(Q_i^k)$ 进行硬判，产生译码结果

$$\hat{z}_i = \begin{cases} 1 & L(Q^k_i) < 0 \\ 0 & L(Q^k_i) \geq 0 \end{cases}$$

将得到的译码结果序列 \hat{Z}^k 左乘校验矩阵, 获得各个校

验式的校验结果: $S^k = (s_1^k, s_2^k, \dots, s_M^k)$ 。

(6)重复2到5, 直到 $S^k = 0$ 或到达最大的迭代次数。

Min-Sum算法迭代的基本结构和BP算法相似, 只是在迭代译码的过程中只有加法和比较大小的运算, 运算量和存储量都要比BP算法小得多, 和适合硬件实现。

4.2.3 Min-Sum译码算法的改进

Min-Sum算法的 (4.8)要比BP算法的 (4.5)所得值偏大, 为了提高Min-Sum译码算法的性能, 将 (4.8)除以一个参数 η ($\eta > 1$), 我们称之为修正最小和算法, 或者加上一个修正值, 称为截短最小和算法, 使之减小与 (4.5)的差距,

第一种修正Min-Sum只需对式(4.8)进行修正, 得到公式:

$$L(r_{ji}) = \left(\prod_{i \in R(j) \setminus i} \alpha_{i,j} \right) \min(\beta_{i,j}) / \eta \quad (4.10)$$

式 (4.10)中 $\beta_{i,j}$ 的最小值除以了一个归一化因子 η , 其中 η 为一个大于1的正数, 通常由三种方法确定 η 的值:

1. 分别定义由对数域BP算法求得的 $L(r_{ji})$ 和Min-Sum算法求得的 $L(r_{ji})$ 为 L_1 和 L_2 ,

$$\text{则 } \eta = \frac{E(L_2)}{E(L_1)}$$

2. 通过一种称为“密度演变” (density evolution) 的算法来近似估计不同归一化因子下归一化算法的噪声门限, 选出其最大噪声门限对应的 η 。

3. 采用计算机仿真的方法确定不同LDPC码的最优归一化因子。

第二种修正Min-Sum只需对式(3.19)进行修正, 得到公式:

$$L(r_{ji}) = \left(\prod_{i \in R(j) \setminus i} \alpha_{i,j} \right) \cdot \max \left(\min |\beta_{i,j}| - \beta, 0 \right) \quad (4.11)$$

相对于规则LDPC码而言, 不规则的LDPC码由于校验节点的度并不一致, 这给 η 值的寻找带来了不少麻烦。如果不同度的校验节点在计算时使用相同的 η 值显然不会的到最佳的性能提高, 但是如何为具有不同度的校验节点找到合适的 η 值缺非常困难。Jun Heo用方法3对一个码长为10000、码率为1/2的规则(3,6)LDPC码做了分析, 找到适合该码的最优 η 值为0.8。改进最小和算法较最小和算法将该码字的性能提高了0.65dB。可以看到改进最小和算法的解码步骤和置信传播算法的相同, 它的运算复杂度和最小和算法几乎一致。如果我们能够找到合适的 η 值, 那么该算法的解码性能将非常逼近置信传播算法。优越的性能和极低的复杂度, 使得改进最小和算法非常适合实际应用。

同Min-Sum算法一样, 改进的Min-Sum译码步骤与对数域BP算法的相同。它的运

算复杂度和Min-Sum算法几乎一样，但如果选取恰当的修正因子，它的译码性能非常逼近BP算法，这使得它成为硬件实现的首选。

4.2.4 LDPC 码其他译码方法

在上面各个章节中，LDPC 码的译码算法都是基于消息传递的译码算法。在 LDPC 码的译码算法中，除了上述基于消息传递的译码算法，还有一类非常重要的基于数学优化的算法，其中有 J.Feldman 提出的基于线性优化的译码算法及 yuheng H 提出的基于协作优化的译码算法

第五章 性能仿真与结果分析

5.1 蒙特卡罗思想

大部分LDPC码的译码都采用软判决。软判决译码算法主要是置信传播(BP, Belief Propagation)算法, 它的译码性能最好。BP算法的译码思想是在Tanner图中的变量节点和校验节点不断的传递概率似然值, 最终实现译码。第一步是更新所有的变量节点的信息;第二步更新所有校验节点的信息。节点的信息更新都是独立的, 因此可以实现并行译码。在对数域的BP译码算法的实现可以大大简化译码计算过程, 并且在此基础上的改进算法可以在性能和算法的复杂度间实现性能的平衡。迭代完成一次, 就进行一次判决, 用校验方程来验证该码字是不是正确的码字, 如果是, 则译码完成得到正确码字, 如果不满足校验方程, 则表明译码算法还未能译出正确码字, 继续信息更新, 直到满足校验方程。如果达到预先设定的最大迭代次数, 还不满足校验方程, 说明译码失败。必须放弃此次译码算法。译码算法详细介绍见第三章。

在一个数字通信系统中, 要确定误码率, 最基本的仿真方法是给系统输入大量的数字符号, 并计算接收机输出端的差错的符号数目, 这种方法叫蒙特卡罗方法^[32]。对一个要估算误码率的系统, 按照蒙特卡罗方法在系统的输入端输入 N 个符号, 在系统输出端记录到 N_e 个差错, 则差错概率的蒙特卡罗估计是

$$\hat{P}_E = \frac{N_e}{N} \quad (5.1)$$

这个估计叫基于 N 个符号的误码率^[32], 给出了符号差错概率的一个估计。使用概率的相对频率定义, 符号差错概率^[32]可定义为

$$\hat{P}_E = \lim_{N \rightarrow \infty} \frac{N_e}{N} \quad (5.2)$$

根据蒙特卡罗思想, 系统仿真的原理框图如图 5.1^[32]:

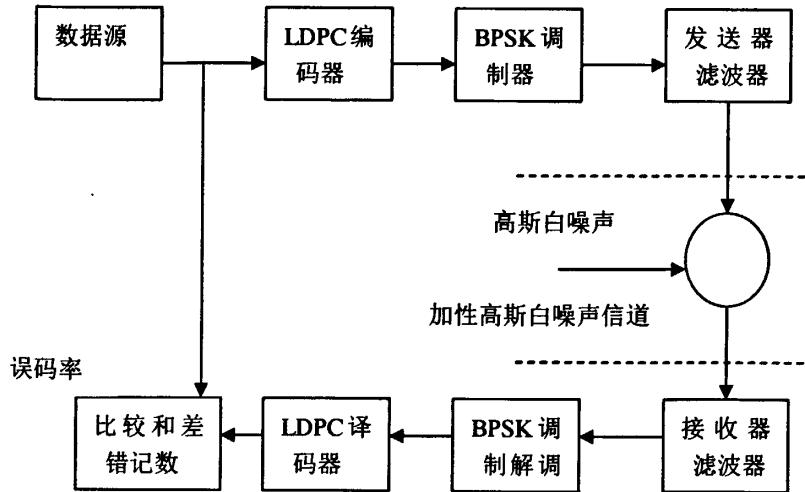


图 5.1 仿真系统框图

针对图 5.1 所示的仿真系统，要想确定和画出在不同信噪比下误码率的变化规律，只需要保持输入信号的功率的恒定，人为的控制噪声功率的变化幅度和变化范围，使信噪比的变化以一定的间距幅度上升。

5.2 性能仿真分析

在前面的几章中我们从理论上对LDPC码的性能进行了，下面我们通过Matlab程序仿真来验证。我们以DVB-S2标准中，码字长度为16200、编码码率为1/4的LDPC码进行仿真。仿真条件是译码采用对数域BP译码算法，迭代次数我们设置它的最大次数为50，对误码率，误帧率进行了探讨。从图5.2的仿真结果来看，码长16200，码率1/4的LDPC码的纠错能力极其的好，信噪比在0.4比特时，误码率为 10^{-1} ，随着信噪比的加大，误码率变得越来越小，并且误码率下降的速度也越快，当信噪比为2dB时，误码率已达到 10^{-6} ；当它与其他编码相结合时，尤其是和BCH码级联时，其性能将更好。

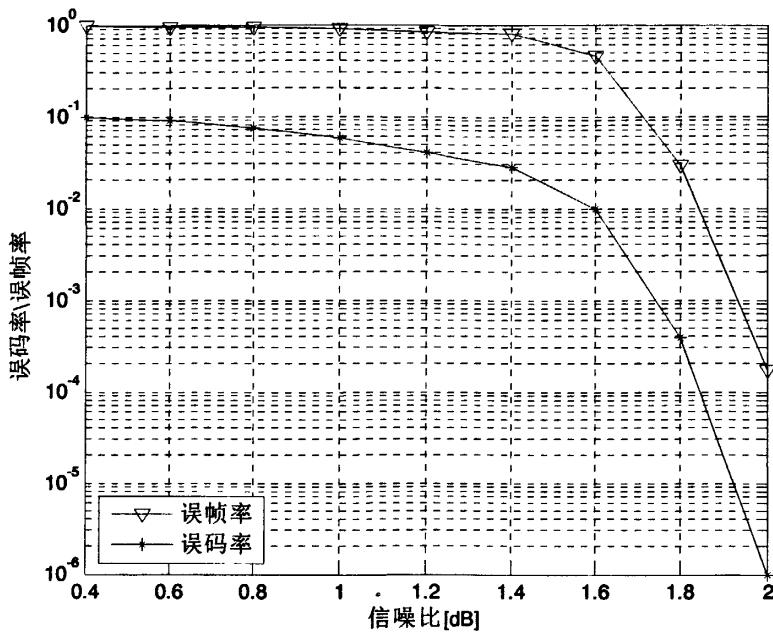


图 5.2 和积算法下的误码率和误帧率

对数域的BP译码算法虽然其性能很好，也改善了标准BP算法中乘法运算较多的缺点，但是我们从第四章也能看出。对数域BP算法即和积算法还是存在较多的和算法，为此提出了最小和算法。图5.3给出了在误码率为 10^{-1} 时，和积算法和最小和算法在不同信噪比条件下的平均迭代次数。

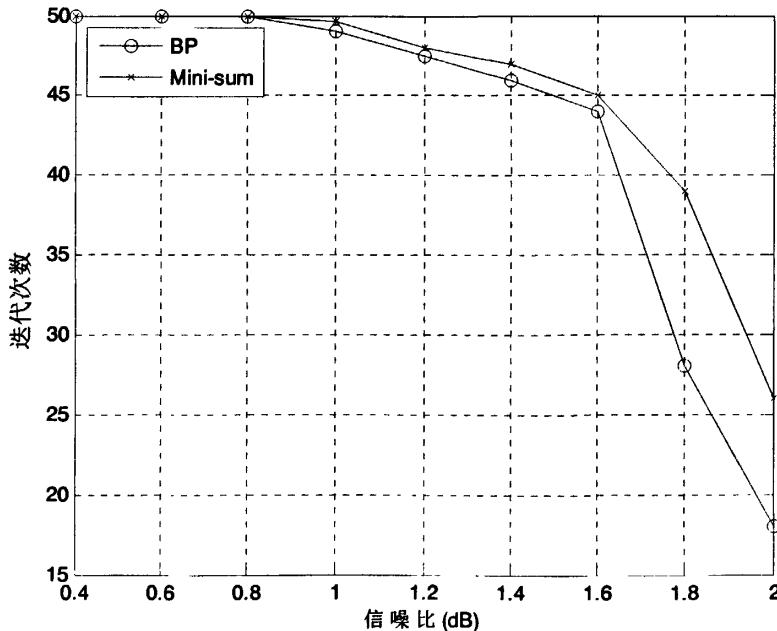


图 5.3 平均迭代次数

从图 5.3 我们可以看出在误码率要求相同的情况下，和积算法和最小和算法的平

均迭代次数都是随着信噪比的增加，迭代次数越来越小。但是和积算法下降的速度更快。

我们采用和积算法和最小和算法对码长为 16200、码率为 1/4 的 LDPC 码进行译码仿真，最大迭代次数依然设为 50 次。结果如图 5.4 所示。

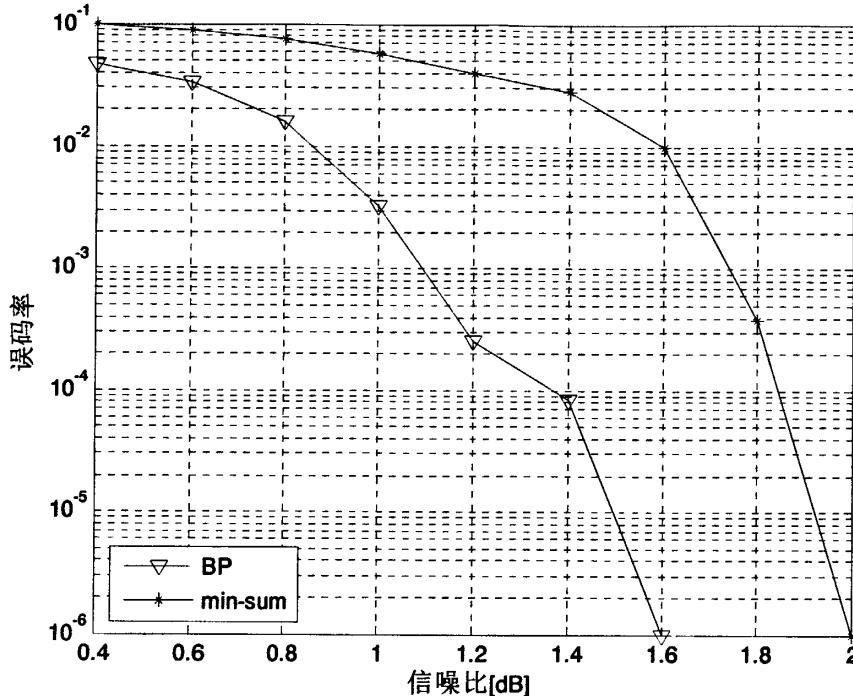


图 5.4 BP 算法和最小和算法性能比较

从图5.4可看出和积算法和最小和算法译码性能都是随着信噪比的加大，误码率快速减小的，但和积算法曲线误码率下降更快。在误码率为 10^{-6} 时，最小和算法比和积算法在性能上降低了大约0.4dB。

由图5.3,5.4可以看出，最小和算法虽然在算法的复杂度上大为减小，但是在性能上的损失也是非常大的。为此我们采用在第三章的译码算法介绍中提到的改进 min-sum 译码算法来达到既减小算法复杂度，又能使性能不会受到大的损失。

5.3 修正因子和偏移因子的选择

在这里我们设定对数域算法的校验节点输出为 L_1 ，而 min-sum 算法输出则为 L_2 ，那么从第四章的证明可以得出以下的结论： L_1 与 L_2 有相同的符号； L_2 的幅值大于 L_1 的幅值即 $|L_2| > |L_1|$ 。从第四章对可以得出，对和积算法的修正当中，用 $\phi(x)$ 取代 $\sum \phi(x)$ 时，明显存在修正过当，这意味着，要对修正过当进行纠正，于是出现了归一化修正和偏移化休整，要提高最小和译码算法的精确度，必须要减少校验节点输出的信息幅

度,一般归一化修正考虑的思路是除以一个大于1的正整数,偏一化休整一般是减去一个正数。同时,为了防止纠正过当,一般都要乘上一个小于1的正系数。下面就是一种为了减小校验节点的输出幅值而提出的纠正方法。

按照上文推道,要达到减少 $|L_2|$ 的幅值的目的,必然要将校验节点的输出信息 $|L_2|$ 除以一个大于1的正整数因子 η ,使过当修正的最小和算法的输出信息接近和积算法的输出信息即 $|L_1| \leftarrow |L_2| / \eta$ 其中 η ^[36]是归一化因子。

$$\eta = \max \left(0.85 \times \frac{E(|L_1|)}{E(|L_2|)}, 1 \right) \quad (5.5)$$

式(5.5)是求归一化因子的一种表达式。不同结构,不同码率的LDPC的归一化因子的构造是不一样的,用Matlab进行仿真可以得到归一化因子和信噪比的关系图。公式中 $E(|L_1|)$ $E(|L_2|)$ 分别为和积算法输出信息幅值 $|L_1|$,最小和算法输出信息幅值 $|L_2|$ 的均值。两算法中输出信息比值 $E(L_2)/E(L_1)$ 的值一般在1.5与5之间。为了防止过纠错,乘以一个小于1的系数0.85^[39]。 $E(|L_1|)$, $E(|L_2|)$ 的计算方法为

$$E(|L_1|) = E \left(2 \cdot \tanh^{-1} \prod_{i=1}^n \tanh \left(\frac{x_i}{2} \right) \right) \quad (5.6)$$

$$E(|L_2|) = E(\min |x_i|) \quad (5.7)$$

和归一化修正的目的一样。为了达到减少最小和算法中输出信息 $|L_2|$ 的幅值的目的,我们可以考虑减去一个正数 β ,使 $|L_2|$ 接近 $|L_1|$,即

$$|L_1| \leftarrow \text{sgn}(L_2) \max(|L_2| - \beta, 0) \quad (5.8)$$

和归一化因子的求法一样, β 的取值也要考虑纠正过当的问题,为此我们在确定 β 值的时候,也要乘以一个小于1的偏移系数。 β ^[39]的取值方法为

$$\beta = 0.95 \times (E(L_2) - E(L_1)) \quad (5.9)$$

$E(|L_1|)$, $E(|L_2|)$ 的计算同样由式5.6和式5.7计算得出。

图5.5为由公式5.5确定的码率 $R = 1/4$,码长为16200的LDPC码的归一化因子和信噪比关系。从图5.5可以看出,归一化因子随着信噪比的加大,是逐渐下降的,而且归一化因子的取值在1到1.5之间。并且随着信噪比的进一步加大。归一化因子的值逐渐接近常数。图5.6为码率 $R = 1/4$,码长为16200的LDPC码的偏移量 β 值与信噪比变化的关系图。由图5.6可以看出,偏移化因子随着信噪比的增加而逐渐下降,当信噪比进一步加大时。偏移化因子可以看出一个常数,它的值在0.3到0.1之间。在仿真时,我们一般都把归一化因子 η 和偏移因子 β 取固定值。这样使仿真更加容易实现。

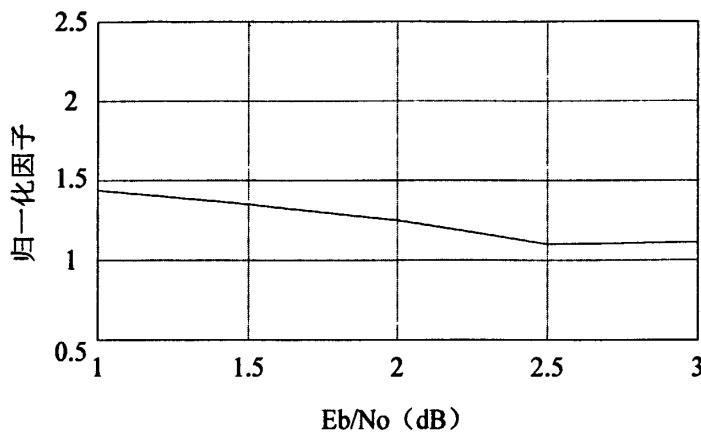


图5.5 归一化因子和信噪比关系

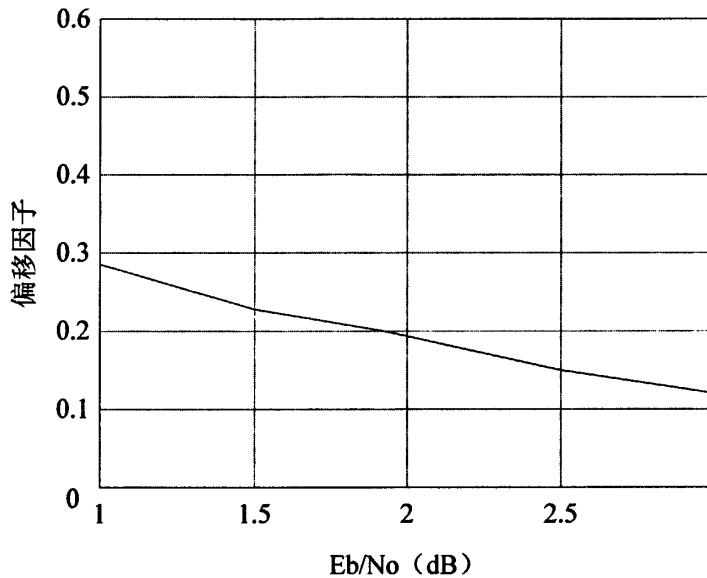


图 5.6 偏移因子与信噪比关系

5.4 不同归一化因子性能分析

根据第四章对 min-sum 算法和归一化 min-sum 译码算法的介绍，我们知道 $\eta > 1$ ，当 $\eta = 1$ 就是 min-sum 算法算法了，针对不同的信噪比设置不同的归一化参数 η 的值可以得到额外的性能提高，但是由于运算过程中 η 值变化不大，所以在一般的计算中 η 可以取一个固定值。我们以 $\eta = 1.175$, $\eta = 1.12$, $\eta = 1.25$, 为例进行仿真。结果如图 5.7 所示。

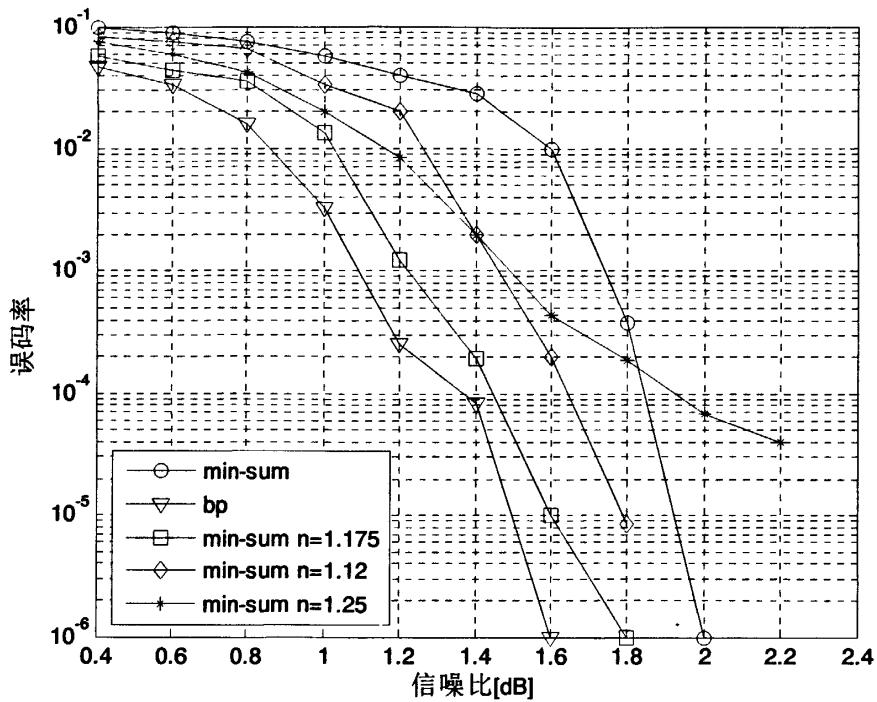


图 5.7 采用改进的 min-sum 算法性能仿真图

从图 5.7 可以看出采用修正了的 min-sum 算法，性能有了很大的改进，当修正参数 η 选取恰当值 ($\eta = 1.175$) 时，与标准 BP 算法相差 0.1dB 以内，选取参数 $\eta = 1.25$ 时，这时产生了误码平台。采用修正的 min-sum 算法能带来简单的硬件实现。

第六章 结论与展望

本文以 DVB-S2 标准中 LDPC 码的应用为研究背景，主要完成了以下几项工作：

(1) 针对一般线性分组码的译码复杂度和码的长度成指数关系，而 LDPC 码由于是有特殊的稀疏校验矩阵，所以它的译码复杂度和码长成线性关系。所以首先重点研究了 LDPC 码的快速编码方法。我们可以利用校验矩阵中信息比特和校验比特的约束关系进行编码，而不必将校验矩阵转化为生成矩阵。根据 DVB-S2 标准中给出的 LDPC 码编码方法，通过仿真得到的非规则校验矩阵，此矩阵可以看成由一个稀疏矩阵和一个双对角线矩阵组成，符合利用校验矩阵直接进行编码的条件，并且编码复杂度为线性。对编码器的给出了初步设计方案，并用 Quartus II 5.1 进行了仿真。

(2) 针对 LDPC 码采用对数域置信传播算法译码时，其校验节点更新运算中包含双曲正切运算，难于实际应用的问题，将几种减小复杂度的译码算法应用于译码中，将复杂函数的运算简化为比较运算，降低了计算复杂度。通过仿真证明，简化译码算法在运算复杂度大幅下降的情况下，其性能与标准译码算法相近。

(3) 利用蒙特卡罗仿真^[32]，利用归一化最小和算法^[39]，取得了比以往方法更好的效果。经仿真验证，归一化算法在设定适合的修正因子的条件下都取得了接近标准置信传播算法的性能。

论文基本达到了预期的研究目标，获得了一定的成果，但是同样存在着很多不足：

(1) 对 DVB-S2 标准中 LDPC 码的译码仅仅作了理论的研究，没有设计出相应的硬件译码器；

(2) 本论文只涉及到了短帧的研究而且只研究了码率为 1/4 的编码，长帧和其他码率的研究将是接下来的研究工作。

致 谢

首先要感谢我的导师陈桂芬教授，在研究生学习期间，陈老师给了我莫大的关心、帮助和教诲，她为我提供了良好的学习条件和宽松的研究氛围，指导我不断地学习、进步。陈老师治学严谨，观察敏锐，思想活跃，这些都深深地影响了我，使我受益匪浅，而她热情真诚、朴实正直、豁达开朗、平易近人，关爱学生的品德更将对我产生久远的影响。

感谢教研室的宋路老师，于林韬老师，朱一锋老师，杨絮老师。他们在课题研究期间给予我很多的指导和帮助，同时他们刻苦钻研的精神也是我学习的榜样。

感谢在教研室和我一起完成课题的师兄，师姐。感谢给我帮助的我的同学们。感谢师弟，师妹的支持和理解。

借此论文完成之际，再一次向所有帮助和支持过我的老师、同学和朋友们表示感激和谢意！我还要特别地感谢我的家人，他们一直给予我最大程度的关怀、爱护、理解和支持。

最后，对评阅本论文的各位专家致以深深的谢意。

参考文献

- [1] C.E.Shannon. "A mathematical theory of communication,"Bell System Technical Journal, July-October 1948
- [2] 王新梅, 肖国镇. 纠错码原理与方法. 西安:西安电子科技大学出版社, 1998, 5:70~97
- [3] Berlekamp,E.R.,McEliece,R.J and van Tilborg,H.C.A.On the intractability of certain coding problems.IEEE Trans.inform.Theory. 1978,24(3): 384~386
- [4] G.D.Forney Jr.,Concatenated Codes.Cambridge,Ma:MIT PRESS.1966
- [5] J.Hagenauer and P.Hoher, "A viterbi algorithm with soft-decision outputs and its applications,"IEEE Trans.Inform.Theory. 1989, IT(13):260~269
- [6] G.Ungerboeck, "Channel coding with multilevel/phase signals," IEEE Trans.Inform.Theory. 1982,28(8):55~67
- [7] C.Berrou and A.Glavieux, "Near Shannon Limit Error-Correcting Coding and Decoding:Turbo-codes," Geneva,Switzerland. 1993,6(4):1064~1070
- [8] C.Berrou and A.Glavieux, "Near optimum error correcting coding and decoding:Turbo-codes,"IEEE Trans. Commun.1996,44:1261~1271
- [9] R.G.Gallager, "Low density parity check codes".Trans.inform.Theory. 1962,IT(8):21~28
- [10] D.J.C.Mackay, "Information Theory.Inference and Learning Algorithms,"Cambridge University Press.2003
- [11] D.J.C.Mackay, "Good error-correcting codes on very sparse matrices,"IEEE Trans Inform Theory. 1999, 45:399~431
- [12] F.R.Kschischang,B.J.Frey, and H.A.Loeliger."Factor graphs and the sum-product algotithm,"IEEE Trans.Inform. Theory.2001,47:498~519
- [13] R.G.Gallager, "Low-Density Parity-Check Codes".Cambridge.MAM.IT Press.1963
- [14] D.Spielman.linear-time encodable and decodable error-correcting codes.IEEE Trans.Inform.Theory. 1996, 42:1723~1731
- [15] V.V.Zyablov.An estimate of the complexity of constructing binary linear cascade codes prob Inform Transm. 1971,7:3~10
- [16] V.V.Zyablov and M.S.Pinsker.Estimtion of error-correction complexity of Gallager low-density codes.Probl Inform Transm. 1976,11:18~28
- [17] M.R.Tanner.A recursive approach to lowcomplexity codes.IEEE Trans.Inform.Theory. 1981,27:533~547
- [18] D.J.C.Mackay and R.M.Neal, "near Shannon limit performance of low-density parity-check codes,"Electron Lett. 1996, 32: 1645~1646
- [19] M.Sipser and D.A.Spielman. "Expander Codes",IEEE Tran. Inform.Theory.1998, 42:1710~1723
- [20] 赵传刚, 林雪红, 林家儒, 吴伟陵. 低密度校验码的研究进展. 电信科学. 2005, (5):48~50
- [21] 孙韶辉, 慕建君, 王新梅. 低密度校验码研究及其新进展. 西安电子科技大学学报(自然科学版). 2001, 28(3):18~21
- [22] 徐澄圻. LDPC 码研究进展. 南京邮电学院学报. 2002, 22 (3):34~38
- [23] T.J.Richardson,R.L.Urbanke.The Capacity of Low-density Parity-Check Codes under Message-Passing Decoding. IEEE Transactions on Information Theory.2001,47(2):599~618

- [24] R.Michael Tanner.A Recursive Approach to Low Complexity Codes.IEEE Transactions on Information Theory. 1981,27(5):533~547
- [25] T.J.Richardson,R.L.Urbanke.Efficient Encoding of Low-Density Parity-Check Codes.IEEE Transactions on Information Theory.2001,47(2):638~656
- [26] Alber to Morello,Vittoria Mignone.DVB-S2:The Second Generation Standard for Satellite Broad-band Services. Proceedings of IEEE.2006,94(1):210~216
- [27] U.Reimers,A.Morello,DVB-S2,the Second Generation Standard for Satellite Broadcasting and Unicasting. Int.J.Satell.Commun.Network.2004,22(3):249~268
- [28] F.Kienle,T.Brack,N.Whern.A Synthesizable IP Core for DVB-S2 LDPC Code.Decoding.Proc.2005 Design, Automation and Test in Europe.Munich,Germany,2005:1~6
- [29] M.Eroz,F.W.Sun,L.N.Lee.DVB-S2 Low-Density Parity-Check Codes with Near Shannon Limit Performance. International Journal on Satellite Communication Networks.2004,22(3):270~278
- [30] F.Kienle,N.Whern.Joint Graph Decoder Design of IRA Codes on Scalable architectures.Proc.2004 Conference on Acoustics,Speech, and Signal Processing.Montreal,Canada.2004:165~169
- [31] 华力,雷箐,于聪梅.DVB-S2 中 LDPC 码编码器的 FPGA 设计与实现. 中国有线电视, 2006 (23)
- [32] 赵健. DVB-S2 标准中低密度校验码的应用研究:[硕士学位论文]. 哈尔滨:哈尔滨工业大学, 2006
- [33] 童胜, 王鹏, 王单, 王新梅. LDPC 码量化和积译码的高效实现. 西安电子科技大学学报(自然科学版). 2004, 31 (5) : 709~711
- [34] Anastasopoulos.A Comparison between the Sum-Product and the Min-Sum Iterative Detection Algorithm Based on Density Evolution.Proc.Globecom Conf.2001:1021~1025
- [35] Jinghu Chen,Ajay Dholakia,Evangelos Eleftheriou.Reduced Complexity Decoding of LDPC Codes.IEEE Transactions on Communication.2005,53(8):1288~1294
- [36] Jinghu Chen,Marc P.C.Fossorier.Near Optimum Universal Belief-propagation Based Decoding of Low-Density Parity-Check Codes.IEEE Transactions on Communication.2002,50(3):406~412
- [37] U.Reimers,A.Morello,DVB-S2,the Second Generation Standard for Satellite Broadcasting and Unicasting. Int.J.Satell.Commun.Network.2004,22(3):249~268
- [38] F.Kienle,T.Brack,N.Whern.A Synthesizable IP Core for DVB-S2 LDPC Code.Decoding.Proc.2005 Design, Automation and Test in Europe.Munich,Germany,2005:1~6
- [39] 高宏峰, 李燕威, 许宗泽. IRA 码最小和译码算法的改进算法. 电光与控制, 2007, 14 (2)
- [40] 于聪梅. 非规则重复积累码的构造研究:[硕士学位论文]. 长沙:国防科学技术大学, 2006