

一种速率匹配的准循环LDPC码的编码构造方法

胡春静 吴湛击 李宗艳 王文博

(北京邮电大学泛网无线通信教育部重点实验室,北京,100876)

摘要:提出了一种新颖的速率匹配的准循环低密度校验码(Low density parity check, LDPC)的编码方法,该方法采用渐进添边(Progressive edge-growth, PEG)算法为工具,综合运用有限几何构造法和搜索方法选取子矩阵的偏移量进行优化,构造LDPC码字,使其生成的校验矩阵具有最短环周期最大化以及短环尽可能少的特点。通过这种方式构造出来的LDPC码,可以实现从 $1/3 \sim 5/6$ 的码率,达到了速率匹配的目的。通过计算机仿真证明,此方法构造的LDPC码字具有良好的误码率和误帧率的性能,并能有效消除LDPC码的“差错地板”现象。并且这种编码方法简单,可节省存储空间,减少编、译码的复杂度。

关键词:低密度校验码;渐进添边算法;准循环码;环长

中图分类号: TN911.22

文献标识码: A

文章编号: 1005-2615(2012)01-0093-07

Construction of Rate-Compatible Quasi-Cyclic LDPC Code

Hu Chunjing, Wu Zhanji, Li Zongyan, Wang Wenbo

(Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, 100876, China)

Abstract: A novel rate-compatible quasi-cyclic low density parity check (LDPC) code is proposed based on progressive edge-growth (PEG) algorithm to optimize the offset of sub matrix through using the finite geometries constructing method and search algorithm. LDPC code has maximum short-ring-cycle and a minimum number of the short rings. Through this method, quasi-cyclic LDPC codes can be obtained with the code-rate range from $1/3$ to $5/6$. Simulation results demonstrate that LDPC code has good frame error ratio and bit error ratio and also can effectively eliminate the error floor phenomenon. Moreover, with low complexity and small storage space, the code is easy to encode.

Key words: low density parity check code; progressive edge-growth algorithm; quasi-cyclic; girth

低密度校验码(Low density parity check, LDPC)是一类采用稀疏的奇偶校验矩阵或随机双向图定义的线性分组纠错码,最初由 Gallager^[1]在1962年提出,1966年D Mackey和R Neal重新发现并证明它在与基于BP的迭代译码相结合的条件下具有逼近香农限的性能,LDPC码开始被广泛地研究。目前,LDPC码在移动通信中得到越来越多的关注,IEEE无线局域网的WiFi标准和国际电联IMT-2000中的Wimax标准都采用了LDPC码,而

在3GPP LTE等下一代移动通信标准中也有很多相关提案。

LDPC码是基于奇偶校验矩阵的,因而如何获得性能优异且尽可能稀疏的奇偶校验矩阵是LDPC编码中的一项关键技术。构造LDPC码的校验矩阵通常分两步:(1)选择度分布表达式;(2)具体构造校验矩阵的结构,即如何安排变量节点和校验节点之间的边的具体放置。已有校验矩阵的构造方法,主要分为代数构造和随机构造两类。代数构造

基金项目:国家重大科技专项基金(2009ZX03003-011-03)资助项目;国家自然科学基金(60702050)资助项目;教育部科学研究重点基金(109013)资助项目。

收稿日期:2010-11-04;**修订日期:**2011-06-08

通讯作者:胡春静,女,副教授,1969年11月生,E-mail:hucj@bupt.edu.cn。

方法,如组合设计方法、有限几何方法等,其优点是能够给出所需校验矩阵的结构,但是由于代数构造方法受限于特定代数结构的特性,无法构造出任意码率的校验矩阵。因此提出随机构造方法,其中 Mackey 等人提出的 MN 构造方法^[2]和 Hu 提出的渐进添边(Progressive edge-growth, PEG)构造方法^[3]是最重要的两种方法。

Mackey 提出的 MN 构造方法的基本思想是:对所构造的 M 行 N 列 LDPC 奇偶校验矩阵,在满足如下条件时可以随机生成,即固定列重,每行的行重尽可能相同,任意两列之间同一位置上的分量都为 1 的情况最多只有一种。Hu 提出的 PEG 方法是一种使得变量节点的局部围长最大的构造方法,其基本思想是通过展开 Tanner 图,找到距离变量节点最远的校验节点,然后在他们之间添加所需要的边,这样,得到的 Tanner 图具有最大的周长,从而可以得到性能较好的 LDPC 码。

自适应编码调制技术在无线移动通信中有着极其重要的作用,因此需要一对编译码就可实现灵活码率编码的速率兼容^[4-6]信道编码方案成为通信的重要选择。本文提出了一种速率兼容 LDPC 码的编码方法,可实现 $1/3 \sim 5/6$ 的码率的灵活变化,得到的码字保持了编码的线性复杂度。

1 优化构造算法

1.1 PEG 算法

PEG 算法具有如下优点:简单,容易构造长码字,并且可以保证周长的下界;便捷,使用密度演化算法优化度序列时,可以生成任意大小、任意码率的好码;略做改变即可用于生成编码时间与码块长度符合线性关系(线性-时间-可编码)的 LDPC 码。但是该算法主要用于随机化 LDPC 码的编码构造,不适合用于结构化 LDPC 码的构造。因此本文提出利用 PEG 算法构造掩码矩阵,这样做的好处是能保证掩码矩阵的周期 ≥ 4 ,且短环数目尽可能少。PEG 算法的基本思想是通过展开 Tanner 图,找到距离变量节点最远的校验节点,然后在他们之间添加所需要的边,这样,得到的 Tanner 图具有最大的周长,从而可以得到性能较好的 LDPC 码。

对于给定的变量节点 s_j ,沿着 s_j 展开深度为 l 的子图,如图 1 所示,所包含的全部校验节点的集合记为 $N_{s_j}^l$,其补集记为 $\bar{N}_{s_j}^l$, $N_{s_j}^l \cup \bar{N}_{s_j}^l = V_c$ 。通过展开 Tanner 图生成一个以变量节点 s_j 为根节点,深度为 l 的子图的步骤如下:

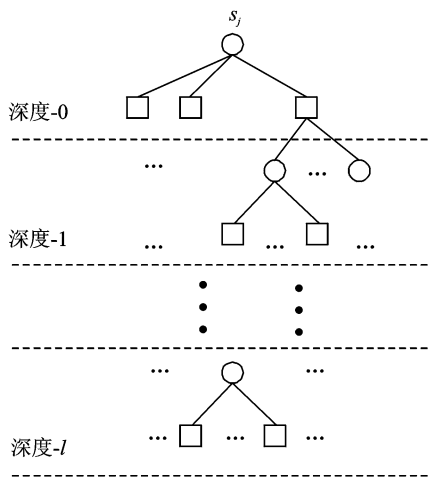


图 1 变量节点 s_j 为根节点,深度为 l 的子图

首先,从变量节点 s_j 开始,首先遍历与其直接相连的校验节点的边 $(s_j, c_{i_1}), (s_j, c_{i_2}), \dots, (s_j, c_{i_{d_{ij}}})$; 然后,从校验节点 $c_{i_1}, c_{i_2}, \dots, c_{i_{d_{ij}}}$ 开始遍历不包含 $(s_j, c_{i_1}), (s_j, c_{i_2}), \dots, (s_j, c_{i_{d_{ij}}})$ 的边,这个过程一直进行下去。考虑到在深度为 l 时,第一次出现在子图中的变量节点到 s_j 的距离为 $2l$,第一次出现在子图中的校验节点到 s_j 的距离为 $2l+1$, $N_{s_j}^l$ 也可以表示为到 s_j 的距离小于或等于 $2l+1$ 的校验节点的集合。同样地,对于给定的校验节点 c_i ,沿着 c_i 展开深度为 l 的子图,所包含的全部变量节点的集合记为 $N_{c_i}^l$,当校验节点二次出现时,它与初始校验节点之间的距离也为周长。

如图 1 所示的子图,存在以下两种情况:

- (1) 集合 $N_{s_j}^l$ 中的元素不再增加,且小于 m ;
- (2) $\bar{N}_{s_j}^l \neq \Phi$, 但 $\bar{N}_{s_j}^{l+1} = \Phi$ 。

对于情况(1),表示尚有不能到达 s_j 的校验节点,所以在这个时候可以添加一条边到这类校验节点且不产生环。对于情况(2),所有的校验节点都能到达 s_j ,则添加一条距离 s_j 最远的边(深度为 $l+1$),新增加的边引入的环具有最大长度,即为 $2(l+1)$ 。

当增加与 s_j 相连的边时,如果出现多个选择,即 $\bar{N}_{s_j}^l$ 有多个校验节点,那么选择当前图中入射边最少的节点。按照这种方法选择校验节点得到的 Tanner 图,其校验节点的度分布是尽可能均匀的。

1.2 偏移量优化算法

准循环 LDPC 码的校验矩阵是一类特殊的校验矩阵,其每一个元素均可以扩展为一个大小为 $z \times z$ 的循环移位矩阵,因此准循环 LDPC 码中偏移量的选择对于此码的好坏起着至关重要的作用。为

此,综合运用有限几何构造法和搜索方法去选取子矩阵的偏移量,可以消去校验矩阵中长度为4和6的短环^[3]。

假设构造出来的校验矩阵为 $\mathbf{H}_b^{\text{uniform}}$,对于校验矩阵 $\mathbf{H}_b^{\text{uniform}}$ 中的每一个元素 $(h_{ij}^b)_{\text{uniform}}$,都需要进行修正处理,以使该修正后的修正矩阵 $\mathbf{H}_b^{\text{modified}}$ 中所有元素的值都小于LDPC码的扩展因子 z ;该变换后的修正矩阵 $\mathbf{H}^{\text{modified}}$ 中第 i 行、第 j 列的元素 $(h_{ij}^b)_{\text{modified}}$ 的计算公式为

$$(h_{ij}^b)_{\text{modified}} = \lfloor (h_{ij}^b) \times \frac{z}{z_{\max}} \rfloor \quad (1)$$

式中: $\lfloor \cdot \rfloor$ 为向下取整的运算符号;扩展因子 $z = \left\lfloor \frac{K}{k_b} \right\rfloor$, $\lceil \cdot \rceil$ 为向上取整的运算符号, K 为信息比特长度, k_b 为对应于信息比特的第一个子矩阵 $\mathbf{H}_b^{\text{system}}$ 中的列重量, $k_b = n_b - m_b$ 。

假设所有循环移位矩阵均不是全零阵,针对循环移位矩阵的特性,有如下结论成立

$$\begin{array}{ccc} p^{h_{1',j_1}^b} & p^{h_{1',j_2}^b} & p^{h_{1',j_3}^b} \\ \left\{ \begin{array}{c} \leftarrow \\ \rightarrow \end{array} \right. & & \\ p^{h_{2',j_1}^b} & p^{h_{2',j_2}^b} & p^{h_{2',j_3}^b} \\ p^{h_{3',j_1}^b} & p^{h_{3',j_2}^b} & p^{h_{3',j_3}^b} \end{array}$$

构造的校验矩阵 \mathbf{H} 所对应的LDPC码不包含长度为4的环的必要条件为

$$(h_{i_1,j_1}^b - h_{i_1,j_2}^b + h_{i_2,j_2}^b - h_{i_2,j_1}^b) \bmod z \neq 0 \quad (2)$$

式中: $h_{i,j}^b$ 为基础矩阵 $\mathbf{H}_b^{\text{uniform}}$ 中的元素, $h_{i,j}^b \in \{0, 1, \dots, z_{\max}\}$, 为基础编码矩阵的循环移位次数。由式(1)可知,基础矩阵 $\mathbf{H}_b^{\text{uniform}}$ 中的每一个不等于-1的元素对于不同码长对应着一个修正后的值,因此式(2)可以被修正为

$$\left(\lfloor h_{i_1,j_1}^b \times \frac{z}{z_{\max}} \rfloor - \lfloor h_{i_1,j_2}^b \times \frac{z}{z_{\max}} \rfloor + \lfloor h_{i_2,j_2}^b \times \frac{z}{z_{\max}} \rfloor - \lfloor h_{i_2,j_1}^b \times \frac{z}{z_{\max}} \rfloor \right) \bmod z \neq 0 \quad (3)$$

定理1 假设码长较长的准循环LDPC码基础矩阵 $\mathbf{H}_b^{\text{uniform}}$ 中任意两对分别同行同列且值不等于-1的元素 $h_{i_1,j_1}^b, h_{i_1,j_2}^b, h_{i_2,j_2}^b, h_{i_2,j_1}^b$, 令 $d = h_{i_1,j_1}^b - h_{i_1,j_2}^b + h_{i_2,j_2}^b - h_{i_2,j_1}^b$, $\omega = \{t \mid \frac{2}{z} z_{\max} < |t| < \frac{z-2}{z} z_{\max} \text{ or } \frac{z+2}{z} z_{\max} \leq |t| \leq \frac{2z-2}{z} z_{\max} \text{ or } \frac{2z-1}{z} z_{\max} < |t| < 2z_{\max}\}$, 如果 $d \in W$, 那么对应的LDPC码不包含长度为4的环。

因此,修正偏移量算法如下:

For $i=1$ to m_b do

Begin

For $j=1$ to n_b do

Begin

遍历矩阵 $\mathbf{H}_b^{\text{uniform}}$ 中的元素,任意选取其中两对分别同行同列且值不等于-1的元素 $h_{i_1,j_1}^b, h_{i_1,j_2}^b, h_{i_2,j_2}^b, h_{i_2,j_1}^b$, 令 $x = h_{i_1,j_1}^b - h_{i_1,j_2}^b + h_{i_2,j_2}^b - h_{i_2,j_1}^b$, 使得 x 满足 $x \in W$, 否则对元素 h_{i_2,j_2}^b 进行修正处理,使其满足以上条件。

End

End

经过偏移量算法修正后的矩阵 $\mathbf{H}^{\text{modified}}$ 即是实际编译码时所需要的矩阵。

2 速率匹配准循环LDPC构造

本节结合打孔模式与扩展模式的优点,提出一种新的速率匹配模式,即是采用打孔和扩展相结合的方式构造速率匹配的LDPC码。因此,LDPC码的实现有3个主要的环节:选取合适的母码码率并构造性能优良的LDPC码,这是确保速率匹配的LDPC码性能的前提条件;在此前提基础上采用扩展模式构造低码率的LDPC码以及采用打孔模式构造高码率的LDPC码。

2.1 母码构造

要实现从1/3~5/6的码率,首先要选择合适的母码码率。单靠打孔的方式不能保证在很大码率范围内LDPC码的性能。一方面,若采用1/3码率作为母码码率,为得到高码率的LDPC码,打孔打掉了大部分译码器需要的软信息,从而致使译码失败,通过仿真实验验证单靠打孔的方式很难保证高码率时LDPC码的性能。另一方面,如果母码码率过高,在信道条件不好的情况下,低码率实现起来比较复杂,造成资源浪费。因此,结合理论分析和仿真实践,最后选定母码码率为1/2。

选定母码码率以后,采用前述的准循环LDPC码的优化构造算法来构造码率为1/2的母码。该方法具体操作步骤如下:

(1)根据码率 R 和码长 N 的要求,确定需要构造的基础矩阵 \mathbf{H}_b 的行数 m_b 和列数 n_b , 并采用密度演化算法确定基础矩阵 \mathbf{H}_b 的列重量分布;其中,码率 R 的计算公式为: $R = K/N$, 校验比特 M 的长度为: $M = N - K$, 其中,信息比特长度 K 、码长 N 和校验比特长度 M 均为正整数。

根据上述讨论结果,选择的母码码率为1/2, $m_b = 16$, $n_b = 32$, 基础矩阵 \mathbf{H}_b 的列重分布为:

$\lambda(x) = 0.25x^2 + 0.25x^3 + 0.125x^4 + 0.0625x^8 + 0.1875x^{16}$, 即列重量为 2, 3, 4, 8, 16 的列数占总列数的比例分别为 0.25, 0.25, 0.125, 0.0625, 0.1875。

(2) 利用 PEG 算法构造一个与基础矩阵 H_b 相对应、且结构相同的原始基础矩阵 H_b^{MASK} : 该原始基础矩阵 H_b^{MASK} 也是一个 m_b 行 n_b 列的矩阵, 且其列重量与基础矩阵 H_b 相同; 该原始基础矩阵 H_b^{MASK} 由对应于信息比特的 m_b 行 m_b 列的第一个子矩阵 A 和对应于校验比特的 m_b 行 m_b 列的第二个子矩阵 B 组成, 即: $H_b^{\text{MASK}} = [A|B]$; 该原始基础矩阵 $H_b^{\text{MASK}} = [A|B]$ 中的元素值为“0”或“1”, 在由原始基础矩阵 H_b^{MASK} 生成基础矩阵 H_b 时, “0”表示没有矩阵运算, “1”表示有矩阵运算。

第二个子矩阵 B 采用的是准双下三角结构, 即该子矩阵第一列的列重量为 3, 其余部分是一个双下三角结构。然后, 根据第二个子矩阵 B 的结构和原始基础矩阵 H_b^{MASK} 的列重量分布, 采用 PEG 算法来构建第一个子矩阵 A , 构成原始基础矩阵 H_b^{MASK} , 如图 2 所示。

(3) 根据原始基础矩阵 H_b^{MASK} 生成基础矩阵 H_b : 先将“-1”填充到原始基础矩阵 H_b^{MASK} 中“0”的

```

1 0 0 0 0 0 1 0 0 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0 0 1 0 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 1 0 0 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 1 0 0 0 1 0 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 1 0 1 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
0 1 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
0 0 0 0 0 1 0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
0 0 1 0 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

```

图2 原始基础矩阵 H_b^{MASK}

位置上, 再在原始基础矩阵 H_b^{MASK} 中“1”的位置上配置不同的循环移位矩阵的偏移量, 再利用偏移量优化算法优化得到基础矩阵 H_b 。

对子矩阵 A 和 B 分别配置偏移量。首先, 将“1”填充到 A 和 B 中“0”的位置上; 接着, 对于在第一个子矩阵 A 中, “1”位置上填充的偏移量数值则是通过计算机搜索, 最大限度地随机化选取循环移位矩阵的初始化偏移量

$$h_{ij}^b = [(j+1) \times 10 \times (i+5) + i^2 + i \cdot (j+3)] \bmod (z_{\max})$$

得到系统位子矩阵 H_b^{system} , 其中, $z_{\max} = 565$; 然后, 对在第二个子矩阵 B 中列重量为 3 的那一列元素中, 中间的“1”位置上填充一个正整数, 其余的“1”的位置上均填充“0”, 得到校验位子矩阵 H_b^{parity} 。

基础矩阵 $H_b =$

```

50 -1 -1 -1 -1 -1 350 -1 -1 500 -1 -1 -1 135 185 235 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 125 -1 -1 -1 -1 -1 491 -1 -1 109 -1 -1 292 353 414 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 224 -1 -1 -1 -1 -1 91 -1 -1 307 -1 451 523 30 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 347 -1 -1 31 -1 -1 -1 363 -1 -1 47 130 213 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
118 -1 -1 -1 -1 23 -1 -1 -1 -1 -1 22 -1 210 304 398 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 560 -1 -1 -1 415 -1 60 -1 -1 375 480 20 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 280 -1 -1 -1 -1 -1 -1 78 -1 310 -1 542 93 209 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 6 -1 -1 -1 514 -1 -1 -1 19 146 273 400 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 192 -1 -1 -1 179 317 455 28 135 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1
-1 -1 546 -1 -1 -1 -1 -1 -1 459 -1 -1 341 490 74 223 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1
-1 -1 -1 195 355 -1 -1 -1 -1 -1 345 -1 100 260 420 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1
-1 485 -1 -1 -1 -1 -1 552 -1 -1 -1 106 277 448 54 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1
-1 -1 -1 -1 -1 130 -1 -1 -1 293 -1 -1 274 456 73 255 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1
-1 -1 209 -1 -1 -1 416 -1 -1 -1 -1 444 72 265 458 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1
-1 -1 -1 -1 114 -1 -1 161 -1 -1 -1 -1 51 255 459 98 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0
-1 -1 -1 -1 -1 415 -1 -1 -1 -1 360 -1 225 440 90 305 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0

```

(4) 对基础矩阵 H_b 进行变换处理, 得到变换后的基础矩阵 H'_b , 以使该修正后的基础矩阵 H'_b 中所有元素的值都小于 LDPC 码的扩展因子 z ; 该修正变换后的基础矩阵 H'_b 中第 i 行、第 j 列的元素 $(h'_{ij})'$

的计算公式为: $(h'_{ij})' = \lfloor (h'_{ij}) \times \frac{z}{z_{\max}} \rfloor$, 其中, 扩展因子 $z = \left\lfloor \frac{K}{k_b} \right\rfloor$, K 为信息比特长度, k_b 为对应于信息比特的第一个子矩阵 A 中的列数, $k_b = n_b - m_b$ 。

设信息码长 $K=5114$,扩展因子 $z=\lceil \frac{K}{k_b} \rceil = 320$,此时得到的修正变换后的基础矩阵 H'_b 也是由系统位子矩阵 H_b^{system} 和校验位子矩阵 H_b^{parity} 共同构成,其中

$$H_b^{\text{system}} =$$

```

28 -1 -1 -1 -1 -1 198 -1 -1 283 -1 -1 -1 76 104 133
-1 70 -1 -1 -1 -1 -1 278 -1 -1 61 -1 -1 165 199 234
-1 -1 126 -1 -1 -1 -1 -1 51 -1 -1 173 -1 255 296 16
-1 -1 -1 196 -1 -1 17 -1 -1 -1 205 -1 -1 26 73 120
66 -1 -1 -1 -1 13 -1 -1 -1 -1 12 -1 118 172 225
-1 -1 -1 -1 317 -1 -1 235 -1 33 -1 -1 212 271 11
-1 158 -1 -1 -1 -1 -1 -1 -1 44 -1 175 -1 306 52 118
-1 -1 -1 3 -1 -1 -1 291 -1 -1 -1 -1 10 82 154 226
-1 -1 -1 -1 -1 -1 -1 108 -1 -1 -1 101 179 257 15
-1 -1 309 -1 -1 -1 -1 -1 259 -1 -1 193 277 41 126
-1 -1 -1 110 201 -1 -1 -1 -1 -1 195 -1 56 147 237
-1 274 -1 -1 -1 -1 -1 312 -1 -1 -1 60 156 253 30
-1 -1 -1 -1 -1 73 -1 -1 -1 165 -1 -1 155 258 41 144
-1 -1 118 -1 -1 -1 235 -1 -1 -1 -1 251 40 150 259
-1 -1 -1 -1 64 -1 -1 91 -1 -1 -1 -1 28 144 259 55
-1 -1 -1 -1 -1 235 -1 -1 -1 -1 203 -1 127 249 50 172

```

$$H_b^{\text{parity}} =$$

```

0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0
0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1
101 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1
0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0
0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0

```

(5)对修正变换后的 m_b 行 n_b 列的基础矩阵 H'_b 进行扩展,得到实际编码时所需使用的 $(m_b \times z)$ 行 $(n_b \times z)$ 列的校验矩阵 H 。

2.2 低码率LDPC码的构造

基于上述母码的构造,下面通过扩展模式构造低码率LDPC,从码率为 $1/2$ 的母码扩展为码率为 $1/3$ 的扩展LDPC码。

构造低码率的LDPC码,重点在于减少短环数量,如图3所示。在构造时,将右上角的矩阵置为零矩阵,右下角与下半部分中间的矩阵置为图中所示的结构,能够避免在右下角出现环长为4的环。

具体实施方式如下:

步骤1 根据码率 R 和码长 N 的要求,确定信息比特长度 K 、校验比特长度 M 、选择母码矩阵 H_b 、扩展部分的行列数 Δm 和扩展矩阵的大小。

母码矩阵选择2.1节构造的 H_b ,母码矩阵 H_b 由大小为 16×16 的系统位子矩阵 H_b^{system} 和大小为 16×16 的校验位子矩阵 H_b^{parity} 组成,其中 H_b^{parity} 为双对角线的准下三角矩阵,码率为 $1/3$ 。经过计算,

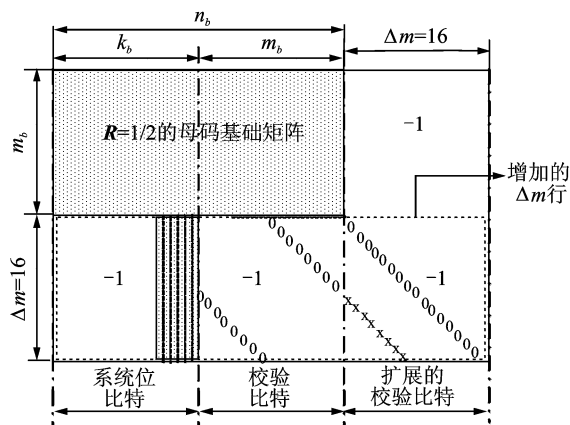


图3 基于扩展模式的基础扩展矩阵结构示意图

扩展部分行列数 Δm 均等于16;因而得出基础扩展矩阵 H_b^{ext} 行数为32,列数为48。

步骤2 基础扩展矩阵 H_b^{ext} 可表示为

$$H_b^{\text{ext}} = \begin{bmatrix} H_b^{\text{system}} & H_b^{\text{parity}} & H_{b^{\text{upper_right}}} \\ H_{b^{\text{lower_left}}} & H_{b^{\text{lower_mid}}} & H_{b^{\text{lower_right}}} \end{bmatrix}$$

式中:母码矩阵 H_b 大小为 $m_b \times n_b$, $H_b = [H_b^{\text{system}} | H_b^{\text{parity}}]$,扩展部分的矩阵 $H_{b^{\text{upper_right}}}$, $H_{b^{\text{lower_left}}}$, $H_{b^{\text{lower_mid}}}$, $H_{b^{\text{lower_right}}}$ 大小分别为 $m_b \times \Delta m$, $\Delta m \times (n_b - m_b)$, $\Delta m \times m_b$, $\Delta m \times \Delta m$,将扩展部分右上角矩阵 $H_{b^{\text{upper_right}}}$ 的元素全部置为“-1”。

如步骤1所述, $\Delta m = 16$,因此基础扩展矩阵 H_b^{ext} 包括5个部分,左上角是大小为 16×32 的母码矩阵 H_b , $H_b = [H_b^{\text{system}} | H_b^{\text{parity}}]$,扩展部分 $H_{b^{\text{upper_right}}}$, $H_{b^{\text{lower_left}}}$, $H_{b^{\text{lower_mid}}}$, $H_{b^{\text{lower_right}}}$ 矩阵大小均为 16×16 。

步骤3 确定 $H_{b^{\text{lower_mid}}}$ 的结构,采用PEG算法构造 $H_{b^{\text{lower_left}}}$ 和 $H_{b^{\text{lower_right}}}$,并采用偏移量优化算法配置偏移量。

为了改善低信噪比条件下扩展LDPC码的性能,在选择左下角矩阵 $H_{b^{\text{lower_left}}}$ 时尽量以增加列重,均匀行重为标准,优化后左下角矩阵的度分布为: $\{0,0,0,0,0,0,0,0,0,0,0,0,2,2,16,16\}$ 。为了改善在高信噪比下扩展LDPC码的性能,右下角矩阵 $H_{b^{\text{lower_right}}}$ 采用双对角线的结构和度分布,度分布为 $\{2,2,2,2,2,2,2,2,1,1,1,1,1,1,1,1\}$,考虑到扩展因子 $z_{\text{max}} = 565$,在选择偏移量(图3 $H_{b^{\text{lower_right}}}$ 中标记为 x 的位置)的时候把偏移量限定在小于565的数值范围内。计算方法如下:

For $i=0$ to $\Delta m - 1$

$$(H_{b^{\text{lower_right}}}^{\text{lower_right}})_{i,i+n_b} = 0$$

End

For $i=0$ to $\Delta m/2 - 1$

$$(H_{b^{\text{lower_right}}}^{\text{lower_right}})_{i+\Delta m/2,j} = [2 \times (15 - \Delta m/2 - i)^2 + 5 \times (15 - i)^2 + 3 \times (15 - \Delta m/2 - i) \times (15 - i) +$$

$$4 \times (15 - \Delta m / 2 - i) + 6 \times (15 - i) \bmod z_{\max}$$

End

2.3 高码率LDPC码的构造

为得到码率为 $2/3 \sim 5/6$ 的码字,可以采用打孔模式^[7]对 $1/2$ 码率的码字的校验位进行打孔。打孔的位置是打孔LDPC码性能的一个关键要素。本文采用分块均匀打孔设计。

分块均匀打孔模式操作步骤如下:首先将通过母码校验矩阵编码的码字平均分为 n_b 块,每块的大小为 z ,依次编号为 $0, 1, \dots, n_b - 1$,其中, $n_b = 32$ 。然后对校验块(编号 $16 \sim 31$)按照打孔样式矢量进行打孔。

按照这种思路,设计大小为 m_b 的打孔样式矢量 P 如下

$$P = [17, 19, 21, 23, 25, 27, 29, 31, 18, 24, 22, 28, 30, 20, 26]$$

定义 $k = \text{floor}(y/z)$, $m = y \bmod z$ 。设计 y 比特的母码打孔位置定义如下:

$$P(0) \times z; (P(0) + 1) \times z - 1$$

(删除第一块 z 比特)

$$P(1) \times z; (P(1) + 1) \times z - 1$$

(删除第二块 z 比特)

$$P(k - 1) \times z; (P(k - 1) + 1) \times z - 1$$

(删除第 k 块 z 比特)

$$P(k) \times z; P(k) \times z + m - 1$$

(删除 m 比特)

按照 P 中的顺序依次去除 i 块 $z \times z$ 大小的码字块($i = P(1), P(2), \dots, P(k - 1)$),最后若第 k 块码字如果不足 z 位,则删除第 $P(k)$ 块的前 m 位比特。

以 $2/3$ 码率为例,如图4所示,空白部分为打掉的数据块。

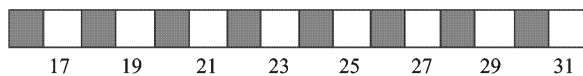


图4 码率为 $2/3$ 的码字的校验位打孔位置

3 性能分析

为了鉴定本文构造的LDPC码的性能,采用了AWGN信道模型对比以下码字的误帧率(FER)性能。

- (1)3GPP中的Turbo码^[8];
- (2)LTE^[7]中的LDPC码,记为LDPC;
- (3)本文提出的LDPC码,记为New-LDPC。

表1给出了仿真参数,仿真结果如图5,6所示。

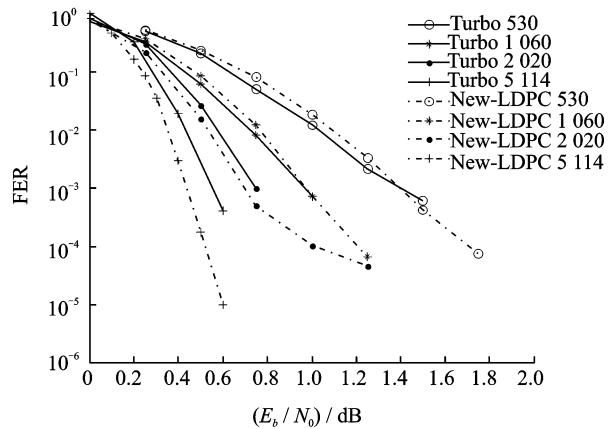


图5 码率为 $1/3$ 时Turbo码与New-LDPC的对比

表1 仿真参数

参数	值
信息位长度	530, 1 060, 2 020, 5 114
码率	$1/3, 1/2, 2/3, 3/4, 5/6$
调制方式	QPSK
信道模型	AWGN, Rayleigh Fading
译码方式	Turbo: Log-MAP MaxIter=8 LDPC: Log-BP MaxIter=50 NEW-LDPC: Log-BP MaxIter=50

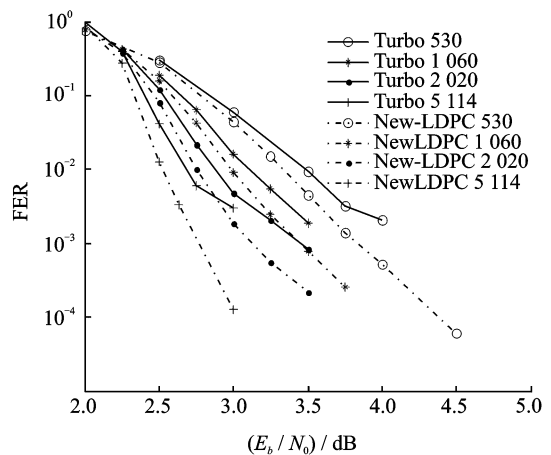


图6 码率为 $4/5$ 时Turbo码与New-LDPC的对比

图5,6显示出本文构造的LDPC码和3GPP中的Turbo码在AWGN信道下 $1/3$ 和 $4/5$ 码率、不同码长的性能对比。从图中可以看出,本文构造的LDPC码的误帧率明显低于Turbo码,在码长较长和码率较高时这种优势更加显著。另外,由于LDPC码能够进行并行译码,因此其吞吐量能够得到大幅提高,硬件复杂度可以大大降低,这些都是LDPC码相对于Turbo码无法比拟的优势。

图7,8为LTE提出的LDPC码与本文构造的LDPC码在AWGN信道下的比较。从图中可以看

出,对于中短码的各个码率,两者的性能相当;而随着码长的增加,本文构造的LDPC码有更低的“差错地板”。

图9为3GPP中的Turbo码与本文构造的LDPC码在Rayleigh衰落信道下的比较。从图中可

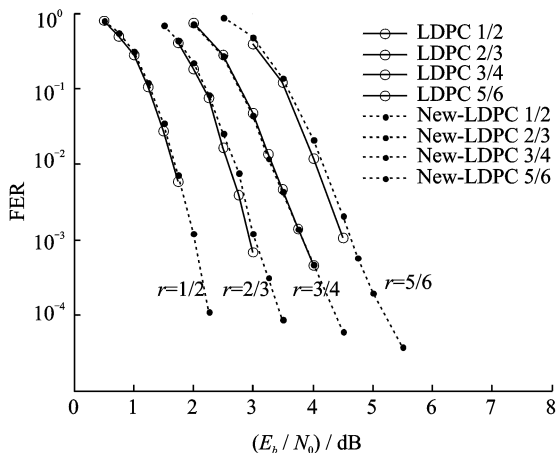


图7 信息位 $k=530$ 时LTE中的LDPC与New-LDPC的对比

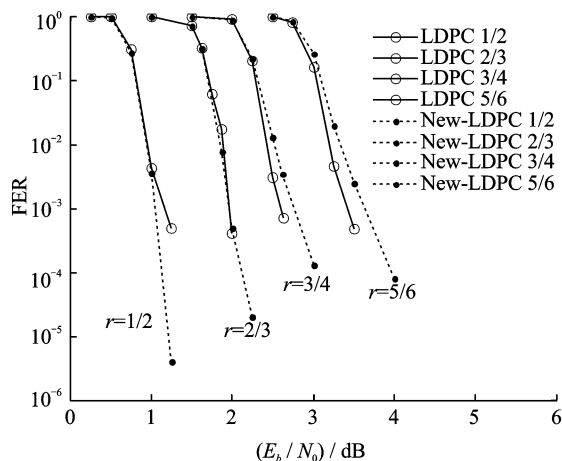


图8 信息位 $k=5114$ 时LTE中的LDPC与New-LDPC的对比

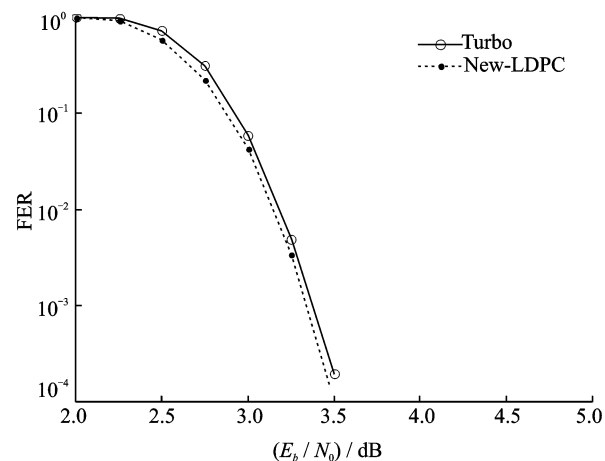


图9 码率为1/2时Turbo码与New-LDPC的对比

以看出,本文构造的LDPC码有更低的“差错地板”。

5 结束语

本文给出了一种基于PEG构造的LDPC码的编码构造方法。为了避免短环对LDPC码性能的影响,综合采用了PEG和偏移量的构造方法减少短环,并通过打孔和扩展,使其达到速率匹配的目的。这种方法构造出来的LDPC码性能优越,在码长很长的时候性能显著优于Turbo码与LTE中的LDPC。另外,由于结构化准循环LDPC码具有较低的编译码复杂度、较高的并行译码吞吐量、更低的“差错地板”和灵活的可扩展性,因而其适用于未来移动通信以及深空通信的编码领域。此外,本文对于码率LDPC码的构造,只是简单地考虑了均匀分块打孔的方法,没有对其他方式进行进一步研究和对比。

参考文献:

- [1] Gallager R G. Low density parity check codes[J]. IRE Transactions on Information Theory, 1962, 8(1):21-28.
- [2] MacKey D J C. Good error correcting codes based on very sparse matrices[J]. IEEE Transactions on Information Theory, 1999, 45(2):399-431.
- [3] Hu Xiaoyu, Eleftheriou E, Arnold D M. Regular and irregular progressive edge-growth tanner graphs [J]. IEEE Transactions on Information Theory, 2005, 5(1):386-398.
- [4] 李森,马林华,杜雅秀.一种正则准循环LDPC码编码器的优化设计[J]. 微计算机信息, 2008, 24(24): 199-201.
Li Sen, Ma Linhua, Du Yaxiu. Optimization design of regular quasi-cyclic LDPC encoder [J]. Microcomputer Information, 2008, 24(24):199-201.
- [5] Kline D, Ha J, McLaughlin S W. On rate-adaptability of nonbinary LDPC codes[C]//5th International Symposium on Turbo Codes and Related Topics. Lausanne: IEEE, 2008:231-236.
- [6] Zhou Lin, Bai Baoming, Xu Ming. Design of nonlinear rate-compatible LDPC codes utilizing bit-wise shorting method[J]. IEEE Communication Letters, 2010, 14(10): 963-965.
- [7] Structured LDPC coding with rate matching [S]. 3GPP TSG RAN WGI #44bis R1-061019. LTE, Athens, Greece, 2006: 27-31.
- [8] Valenti M C, Sun J. Turbo codes (chapter 12 in handbook of RF and wireless technologies) [M]. Edited by Dowla F. [S. l.]: Newnes, 2004: 375-399.