



(12) 发明专利

(10) 授权公告号 CN 102694625 B

(45) 授权公告日 2014. 11. 12

(21) 申请号 201210202279. 2

审查员 刘锐

(22) 申请日 2012. 06. 15

(73) 专利权人 北京邮电大学

地址 100876 北京市海淀区西土城路 10 号

(72) 发明人 牛凯 陈凯

(74) 专利代理机构 北京德琦知识产权代理有限公司 11018

代理人 夏宪富

(51) Int. Cl.

H04L 1/00 (2006. 01)

(56) 对比文件

CN 102122966 A, 2011. 07. 13, 全文 .

CN 102164025 A, 2011. 08. 24, 全文 .

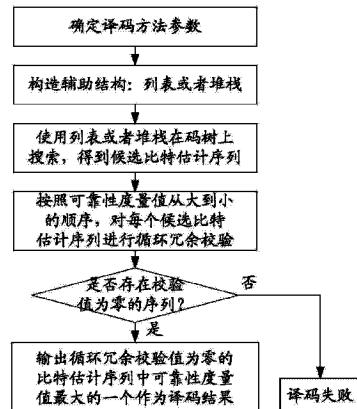
权利要求书4页 说明书9页 附图9页

(54) 发明名称

一种循环冗余校验辅助的极化码译码方法

(57) 摘要

一种循环冗余校验辅助的极化码译码方法，是对极化码进行译码时，在极化码所对应的码树上，从根节点到叶子节点、且其所对应比特估计序列的循环冗余校验值为零的所有路径中，采用列表或堆栈作为路径搜索的辅助，寻找其中可靠性度量值最大的一条路径，并输出该路径对应的比特估计序列作为译码结果。该方法的操作步骤包括：根据搜索辅助方法确定参数，构造译码方法的辅助结构，搜索候选比特估计序列和执行循环冗余校验。本发明能大大提高采用极化码作为信道编码的通信系统的纠错能力，且其操作步骤比较简单，运算复杂度与 WCDMA 系统中使用 Turbo 码的编译码方法相当、甚至更低，因此该方法具有很好的实用化前景。



1. 一种循环冗余校验辅助的极化码译码方法,其特征在于:对极化码进行译码时,在极化码所对应的码树上,从根节点到叶子节点、且其所对应比特估计序列的循环冗余校验值为零的所有路径中,采用列表或堆栈作为路径搜索的辅助,寻找其中可靠性度量值最大的一条路径,并输出该路径对应的比特估计序列作为译码结果;所述方法包括以下操作步骤:

(1) 根据搜索辅助方法确定参数:对码长为N的极化码进行译码,设定搜索宽度为L,并确定使用列表还是堆栈进行路径搜索;若使用堆栈译码,还需设定堆栈最大深度D;其中,N、L、D都是自然数;

(2) 构造译码方法的辅助结构,包括下述两种操作内容:

若选用列表译码,则构造一个包含 $2 \times L$ 条路径记录的列表,该列表中的每条路径记录分别用于存储一个最大长度为N的比特估计序列及其对应的可靠性度量值;

若选用堆栈译码,则构造一个最大深度为D的有序堆栈,该堆栈中的每条记录用于存储一个最大长度为N的比特估计序列及其对应的可靠性度量值;且堆栈中的记录是根据可靠性度量值的数值从大到小顺序排列,即位于堆栈顶部的记录的可靠性度量值最大;再构造用自然数对其进行编号排序的N个计数器;

(3) 搜索候选比特估计序列:使用列表或堆栈在码树上进行搜索,得到可靠性度量值最大的L条互不相同的从根节点到叶子节点的长度为N的路径,这L条路径所对应的比特估计序列即为L个候选比特估计序列;

(4) 执行循环冗余校验:按照可靠性度量值从大到小的顺序,对每个候选比特估计序列进行循环冗余校验,一旦遇到某个比特估计序列的校验值为零时,就停止译码,并输出该比特估计序列作为译码结果;如果对步骤(3)得到的L个候选比特估计序列进行循环冗余校验的过程中没有校验值为零,则译码失败。

2. 根据权利要求1所述的方法,其特征在于:所述极化码的码树结构如下:一个码长为N的极化码所对应的码树是一棵由 $(2^{N+1}-1)$ 个节点和 $(2^{N+1}-2)$ 条边组成的满二叉树,其中除去叶子节点以外的每个节点,与其左、右两侧两个后继节点之间的边都分别被标记为0和1,并对该码树中的边根据其到根节点所需经过的节点数进行分层,故序号为d的层是由 $2^d$ 条不同的边所组成;其中,自然数d的取值范围为1到N,即直接连接根节点的边的层序号d=1,而连接叶子节点的边的层序号d=N;从序号d=1的层到序号d=i的层,分别逐层选取一条边,就形成一条包含i条边的路径,再将该路径中各边的标记0或1按照层序号从小到大的顺序排列,就能得到一个长度为 $1 = i$ 的比特估计序列( $\hat{u}_1, \hat{u}_2, \dots, \hat{u}_i$ );

为方便表示,使用符号 $x_1^N$ 表示实数序列或二进制的比特序列( $x_1, x_2, \dots, x_N$ ),再用符号 $x_s^t$ 表示 $x_1^N$ 的一个子序列( $x_s, x_{s+1}, \dots, x_{t-1}, x_t$ ),即用下标s和上标t分别表示子序列中第一个元素和最后一个元素在其原序列中的序号,其中 $1 \leq s \leq N, 1 \leq t \leq N$ ;且当 $s > t$ 时, $x_s^t$ 表示一个空序列,记作 $\phi$ ,空序列的长度 $1 = 0$ ;还设置 $x_1^N$ 的子序列 $x_s^t$ 中分别由奇数序号的元素所构成的奇数子序列为 $x_{s,o}^t$ ,以及由偶数序号的元素所构成的偶数子序列为 $x_{s,e}^t$ 。

3. 根据权利要求2所述的方法,其特征在于:当设定固定比特序列为一个全零序列时,

对于给定的接收信号  $y_1^N$ , 其长度为  $l = i$  的比特估计序列  $\hat{u}_1^i$  的可靠性度量值  $R(\hat{u}_1^i | y_1^N)$  的计算方法有下述三种:

第 1 种: 比特估计序列为空序列  $\phi$  时, 其可靠性度量值  $R(\phi | y_1^N) = 0$ ;

第 2 种: 若序号为  $i$  的比特属于信息比特时, 其可靠性度量值的计算公式为:

$$R(\hat{u}_1^i | y_1^N) = M_N^{(i)}(\hat{u}_1^i | y_1^N) + f(\hat{u}_1^i); \text{ 式中, } M_N^{(i)}(\hat{u}_1^i | y_1^N) = \ln \left( \frac{W_N^{(i)}(y_1^N, u_1^{i-1} = \hat{u}_1^{i-1} | u_i = \hat{u}_i)}{\sum_{v_1^{i-1} \in \mathcal{X}^{i-1}} W_N^{(i)}(y_1^N, u_1^{i-1} = v_1^{i-1} | u_i = v_i)} \right),$$

集合  $\mathcal{X} = \{0, 1\}$ ,  $\mathcal{X}^k$  表示  $k$  个  $\mathcal{X}$  作笛卡尔积所得到的集合,  $W_N^{(i)}(y_1^N, u_1^{i-1} | u_i)$  为序号为  $i$  的比特所对应的极化信道的转移概率函数, 即发送信号  $u_i$  通过极化信道  $W_N^{(i)}$  得到输出  $y_1^N$  和  $u_1^{i-1}$  的概率,  $\ln(\cdot)$  表示以自然对数  $e$  为底的对数函数; 函数  $f(\hat{u}_1^i)$  的取值规则是: 当序列  $\hat{u}_1^i$  中属于固定比特的部分取值都为零时,  $f(\hat{u}_1^i) = 0$ , 否则,  $f(\hat{u}_1^i) = -\infty$ ;  $\infty$  为无穷大的实数;

第 3 种: 若序号为  $i$  的比特属于固定比特时, 其可靠性度量值的计算公式为:

$$R(\hat{u}_1^{i-1}, \hat{u}_i = 0 | y_1^N) = R(\hat{u}_1^{i-1} | y_1^N), \quad R(\hat{u}_1^{i-1}, \hat{u}_i = 1 | y_1^N) = -\infty.$$

4. 根据权利要求 3 所述的方法, 其特征在于: 所述可靠性度量值中的  $M_N^{(i)}(\hat{u}_1^i | y_1^N)$  的计算方法是利用信道极化的递归结构并按照下述递归运算进行的:

当  $i$  为奇数时, 令  $i = 2k-1$ ,

$$\begin{aligned} M_N^{(i)}(u_1^i = \hat{u}_1^i | y_1^N) &= M_N^{(2k-1)}(u_1^{2k-1} = \hat{u}_1^{2k-1} | y_1^N) \\ &= \max^* \left\{ \left[ M_{N/2}^{(k)}(u_k = \hat{u}_{2k-1}, u_1^{k-1} = \hat{u}_{1,o}^{2k-2} \oplus \hat{u}_{1,e}^{2k-2} | y_1^{N/2}) + M_{N/2}^{(k)}(u_k = 0, u_1^{k-1} = \hat{u}_{1,e}^{2k-2} | y_1^{N/2+1}) \right], \right. \\ &\quad \left. \left[ M_{N/2}^{(k)}(u_k = \hat{u}_{2k-1} \oplus 1, u_1^{k-1} = \hat{u}_{1,o}^{2k-2} \oplus \hat{u}_{1,e}^{2k-2} | y_1^{N/2}) + M_{N/2}^{(k)}(u_k = 1, u_1^{k-1} = \hat{u}_{1,e}^{2k-2} | y_1^{N/2+1}) \right] \right\} \end{aligned}$$

当  $i$  为偶数时, 令  $i = 2k$ ,

$$M_N^{(i)}(u_1^i = \hat{u}_1^i | y_1^N) = M_N^{(2k)}(u_1^{2k} = \hat{u}_1^{2k} | y_1^N) = M_{N/2}^{(k)}(u_1^k = \hat{u}_{1,o}^{2k} \oplus \hat{u}_{1,e}^{2k} | y_1^{N/2}) + M_{N/2}^{(k)}(u_1^k = \hat{u}_{1,e}^{2k} | y_1^{N/2+1});$$

其中, 运算符号  $\oplus$  表示模二加操作,  $u_1^N \oplus v_1^N$  表示对这两个序列逐比特做模二加操作, 即  $u_1^N \oplus v_1^N = (u_1 \oplus v_1, u_2 \oplus v_2, \dots, u_N \oplus v_N)$ ;  $\max^*(a, b)$  表示雅可比对数式, 即  $\max^*(a, b) = \ln(e^a + e^b) = \max(a, b) + \ln(1 + e^{-|a-b|})$ ,  $\max(a, b)$  表示选取  $a, b$  中的较大值, 式中  $a$  和  $b$  均为实数; 因计算  $\max^*(a, b)$  需要的计算资源多, 且有对数和指数运算, 硬件不易实现; 故预先为函数  $g(c) = \ln(1 + e^{-c})$  设置一个表, 其定义域为大于或等于零的实数, 即  $c \geq 0$ ; 这样计算  $\max^*(a, b) = \max(a, b) + g(|a-b|)$  时, 其中的  $g(|a-b|)$  就直接查表; 且为继续降低复杂度, 能直接用  $\max(a, b)$  代替  $\max^*(a, b)$  操作, 但这样会引入一定程度的性能损失。

5. 根据权利要求 1 所述的方法, 其特征在于: 所述步骤(3)使用列表的候选比特估计序列进行搜索译码时, 首先执行下述初始化操作:

清空列表后, 向该列表中添加一条其内容为一个长度为零的空序列  $\phi$  的记录, 且设置

其可靠性度量值为零。

6. 根据权利要求 5 所述的方法,其特征在于:所述步骤(3)使用列表的候选比特估计序列进行搜索译码时,在初始化操作后,执行下列操作内容:

(3A) 扩展列表中的候选比特估计序列:此时列表中各个记录的比特估计序列具有相等的长度 1,逐一取出列表中的各条记录,并在被取出的每条记录中的比特估计序列后分别添加一个 0 比特或一个 1 比特,得到两个新的长度为 1+1 的比特估计序列;接着分别计算这两个长度为 1+1 的比特估计序列的可靠性度量值,并以该两个新的比特估计序列及其对应的可靠性度量值为内容生成两条记录;待列表中所有记录被全部取出后,将新产生的记录存入列表中;

(3B) 列表中候选比特估计序列进行竞争:完成步骤(3A)操作后,如果列表中的记录数量小于或等于 L,则直接跳过本步骤而执行步骤(3C);如果列表中的记录数量大于 L,则保留可靠性度量值最大的 L 项,并删除其余各项;

(3C) 判断是否停止搜索:判断此时列表中记录中的比特估计序列长度是否小于 N,若是,则返回继续重新执行步骤(3A);否则,此时列表中各个记录中的比特估计序列即为步骤(4)所需要的一个候选比特估计序列,结束步骤(3)的操作,执行后续步骤(4)。

7. 根据权利要求 1 所述的方法,其特征在于:所述步骤(3)使用堆栈的候选比特估计序列进行搜索译码时,首先执行下述初始化操作:

用非负整数序列  $q_1^N = (q_1, q_2, \dots, q_N)$  中的各个元素分别表示各计数器的数值,并设其为一个全零序列,在清空堆栈后;向堆栈中添加一条其内容为一个长度为零的空序列  $\Phi$  的记录,且设置其可靠性度量值为零。

8. 根据权利要求 7 所述的方法,其特征在于:所述步骤(3)使用堆栈的候选比特估计序列进行搜索译码时,在初始化操作后,执行下列操作内容:

(3a) 执行弹栈操作:将堆栈顶部的记录弹出,查得该记录的所对应的比特估计序列长度为 1,若 1 不为 0,就将序号为 1 的计数器的数值  $q_1$  加 1;

(3b) 栈中候选比特估计序列进行竞争:如果序号为 1 的计数器的数值已经达到了 L,即  $q_1 = L$ ,则检查堆栈中的所有记录,并删除所有比特估计序列长度小于或等于 1 的记录;然后,判断堆栈最大深度减去当前堆栈中的记录数量之差是否大于或等于 2,如果是,则删除当前位于堆栈底部的那条记录;否则,执行后续步骤(3c);

(3c) 扩展堆栈中的候选比特估计序列:在步骤(3a)中被弹出记录中的比特估计序列后添加一个 0 比特或一个 1 比特,得到两个新的长度为 1+1 的比特估计序列,接着分别计算这两个长度为 1+1 的比特估计序列的可靠性度量值;

(3d) 执行压栈操作:以步骤(3c)中得到的长度为 1+1 的两个新比特估计序列及其各自的可靠性度量值为内容,生成两条新记录,并压入堆栈中;再按照可靠性度量值从大到小的顺序重新排列堆栈中的记录,以使位于堆栈顶部的记录具有最大可靠性度量值;

(3e) 判断是否停止搜索:判断此时堆栈中是否已经没有任何记录,若是,则结束该步骤(3)的操作,执行后续步骤(4);否则,再判断位于堆栈顶部的记录中的比特估计序列的长度是否为 N,若是,则将该记录弹出堆栈,该记录中存储的比特估计序列作为步骤(4)所需要的一个候选比特估计序列,并使序号为 N 的计数器的数值  $q_N$  加 1 后,接着判断该序号为 N 的计数器数值  $q_N$  是否已经等于或大于 L,若是,则结束该步骤(3)的操作,执行后续步

骤 (4) ;否则,返回执行步骤 (3a)。

9. 根据权利要求 1 所述的方法,其特征在于 :所述步骤 (3) 中,采用列表译码时,L 个候选比特估计序列是同时得到的;而采用堆栈译码时, L 个候选比特估计序列是按照可靠性度量值从大到小的顺序逐一分别得到的。

10. 根据权利要求 8 所述的方法,其特征在于 :采用堆栈译码方法时,能够将路径搜索步骤中的 (3e) 和步骤 (4) 的循环冗余校验结合起来,使得路径搜索过程能够提前结束,以降低计算复杂度;此时,执行的操作内容是循环冗余校验辅助下比特估计序列的选择;若堆栈中已经没有任何记录,就停止译码,表明译码失败;如果位于堆栈顶部的记录中的比特估计序列的长度为 N,则将该记录弹出堆栈,并使序号为 N 的计数器的数值  $q_N$  加 1;然后,对该比特估计序列进行循环冗余校验;若校验值为零,则停止译码并输出该比特估计序列作为译码结果;若校验值不为零,且  $q_N$  已等于或大于 L,则停止译码,表明译码失败;否则,返回执行步骤 (3a)。

## 一种循环冗余校验辅助的极化码译码方法

### 技术领域

[0001] 本发明涉及一种循环冗余校验辅助的极化码译码方法,用于解决数字通信系统中由于信道对通信过程的干扰,使得传输数据出现差错的问题;确切地说,涉及一种采用极化码作为纠错编码的实际数字通信系统中,通过利用循环冗余校验信息来辅助译码的方法;属于数字通信的信道编码技术领域。

### 背景技术

[0002] 循环冗余校验是一种信道检错方法,它在实际数字通信系统中已经得到了非常广泛地使用。参见图1,在发送数据前,消息发送端首先使用循环冗余校验模块对原始的消息比特序列进行校验计算,得到一个校验比特序列。循环冗余校验比特序列和原始的消息比特一起组成了信道编码器的输入比特。信道编码器的输出比特被送入信道。经过信道传输后,得到一系列接收信号。然后,经过信道译码后的比特序列被送到消息接收端,由消息接收端对该比特序列进行循环冗余校验操作。如果计算得到的校验值为零,就认为信道译码得到的比特序列是正确的。否则,即校验值不为零,则说明接收到的比特序列中包含了一定数量的错误。对于这种情况,实际系统中的消息接收端往往通过反馈信道请求消息,要求发送端重新发送一次该原始的消息比特序列。

[0003] 对于采用迭代译码方法进行信道译码的系统,可以将循环冗余校验与迭代信道译码过程相结合,即在每次迭代后,都对当前迭代所得到的比特序列进行一次循环冗余校验;且一旦在校验值为零时,就让信道译码器提前结束迭代过程,以减低译码复杂度。

[0004] 一个实际应用事例是在WCDMA系统中,对Turbo码进行迭代译码时,就是采用这种方法来节省译码过程中的操作步骤,以减少译码时间。本发明则是针对信道编码为极化码的情况,设计并实现了两种循环冗余校验辅助的译码方法。

[0005] 先介绍极化码:极化码(Polar Codes)是2009年由E. Arikan提出的一种被严格证明可以达到信道容量的构造性的信道编码方法。在进行极化编码之前,首先要对 $N = 2^n$ 个独立的二进制输入信道(或先后N次反复使用同一个信道,即一个信道的N个可用时隙),再利用图2所示的信道极化的基本单元对二进制输入离散信道反复进行极化,式中,n为自然数。最基本的信道极化是对两个相同的未经极化的信道 $W: \mathcal{X} \rightarrow \mathcal{Y}$ 进行单步极化操作,其中, $\mathcal{X}$ 是信道输入符号的集合,对于二进制输入信道, $\mathcal{X}$ 的取值范围为{0, 1}; $\mathcal{Y}$ 是信道输出符号的集合。如图2所示,该极化信道的输入比特分别标记为 $u_1$ 和 $u_2$ ,这两个输入比特通过一个模二加法器输出得到 $x_1$ ,同时将 $u_2$ 直接赋值给 $x_2$ ,即 $x_1 = u_1 \oplus u_2$ , $x_2 = u_2$ , $\oplus$ 为模二加运算。把 $x_1$ 和 $x_2$ 分别送入未经极化信道 $W$ ,其输出为 $y_1$ 和 $y_2$ 。从该信道极化基本单元的输入( $u_1$ 和 $u_2$ )和两个信道的输出( $y_1$ 和 $y_2$ )看,原本独立的两个未经极化的信道 $W$ 被合并成一个两输入两输出的向量信道 $W_2: \mathcal{X}^2 \rightarrow \mathcal{Y}^2$ ,其中 $\mathcal{X}^2 = \mathcal{X} \otimes \mathcal{X}$ ,运算 $\otimes$ 为笛卡尔积。该向量信道包含两个子信道 $W_2^{(1)}: \mathcal{X} \rightarrow \mathcal{Y} \otimes \mathcal{X}$ (输入为 $u_1$ 输出为 $y_1y_2$ )和 $W_2^{(2)}: \mathcal{X} \rightarrow \mathcal{Y}$ (输入为 $u_2$ 输出为 $y_1y_2u_1$ ),这两个子信道即是两个极化信道。经过该单步极化过程,从信道

容量上看,  $I(W_2^{(1)}) + I(W_2^{(2)}) = 2 \times I(W)$ ,  $I(W_2^{(1)}) \leq I(W) \leq I(W_2^{(2)})$ , 其中  $I(\cdot)$  表示求信道容量的函数。也就是说:单步极化后,在和容量保持不变的情况下,相比原本未经极化的信道,极化后的信道容量发生了偏离:一个增加,一个减少。如果对两组已经完成一次极化操作的信道,再在两组互相独立的转移概率相同的极化信道之间,分别进行单步极化操作,该偏离会更加明显,这一组单步极化操作被称为第二层极化操作,而前一组单步极化操作则称为第一层极化操作。每多做一层极化操作,需要的信道数就会比原先多一倍。因此,对  $N = 2^n$  个信道进行完全的极化,共需要  $n$  层极化操作,且每一层的极化操作包括了  $N$  次单步极化操作。如不加特殊说明,“对  $N$  个信道进行极化操作”是指完全极化。

[0006] 理论上已证明,对接近无穷多个信道进行极化操作后,会出现一部分信道的容量为 1(即通过其传输的比特肯定会被正确接收),其余信道容量为 0(即完全无法在其上可靠地传输比特)的现象,而容量为 1 的信道占全部信道的比例正好为原二进制输入离散信道的容量。

[0007] 参见图 3,介绍一个实用的信道极化装置的递归结构:长度为  $N$ (对  $N$  个信道进行极化)的信道极化装置可以用长度为  $\frac{N}{2}$  的信道极化装置作递归操作来表示,递归过程中的

最小单元(即当  $N = 2$  时)就是图 2 所示的基本单元。图 3 中的信道极化装置中有一个长度为  $N$  的比特反转交织器,其功能是:先将输入端的十进制序号  $i$  按二进制表示为  $(b_{n-1} b_{n-2} \dots b_0)$ ,其中,  $n = \log_2 N$ ,再将该二进制序列反序而得到  $(b_0 b_1 \dots b_{n-1})$ ,最后重新按十进制表示成  $\pi(i)$ ,并作为输入序号  $i$  对应的输出序号。比特反转交织器的作用是将输入端序号为  $i$  的比特映射到序号  $\pi(i)$  处。根据编码速率( $R$ )对  $N$  个信道进行极化,并选取其中容量最大的  $K$  个信道(或者等价地,选取可靠性最高的  $K$  个信道;可靠性度量是采用密度进化(Density Evolution)工具或者计算巴塔恰里亚(Bhattacharyya)参数得到的数值),以承载用于传输消息的比特,称该部分比特为信息比特,并称该部分信道为信息信道(其中  $K = \lfloor N \cdot R \rfloor$ ,  $\lfloor \cdot \rfloor$  为向下取整运算),其余未被选中的信道则传输一个约定的比特序列,称其为固定比特序列,并称该部分信道为固定信道(若信道对称则可简单地传输全零序列),从而形成一个从承载信息的  $K$  个比特到最终送入信道的  $N$  个比特的映射关系,这样的一种映射关系即为极化码,码长(编码后得到的二进制信号所包含的比特数)等于信道极化装置的长度  $N$ 。

[0008] 由信息比特和固定比特组成的、送入信道极化装置的二进制信号序列  $(u_1 \dots u_N)$  为编码码块,该编码码块的顺序与其送入的极化信道的序号一致,即  $u_i$  送入  $W_N^{(i)}$ ,其中序号  $i$  是 1 到  $N$  的自然数,  $W_N^{(i)}$  表示将  $N$  个信道  $W$  极化后得到的序号为  $i$  的极化信道。编码码块经过信道极化装置后,得到的信号序列  $x_1 \dots x_N$ ,再通过  $N$  个独立信道  $W$  传输后,接收到的信号序列为  $(y_1 \dots y_N)$ 。译码器的任务就是根据接收信号序列  $(y_1 \dots y_N)$  得到发送比特序列  $(u_1 \dots u_N)$  的一组比特估计值  $(\hat{u}_1 \dots \hat{u}_N)$ 。

[0009] 极化码可以使用串行抵消译码方法:对编码码块中的每个比特按序号  $i$  顺序地从 1 到  $N$  依次进行判决,还可以将串行抵消译码方法描述为一个码树上的搜索过程(参见图 4

所示的一个简单范例)。串行抵消译码是在码树上逐步扩展,每次从两条候选路径中选择其中概率值相对较大的一条,并在那一条路径的基础上继续进行下一步的路径扩展。

[0010] 作为串行抵消译码的一种改良方法:串行抵消列表译码允许将候选路径保留多条,而不是仅仅一条,以此扩大搜索的范围和减小在搜索过程中离开正确路径的概率。所有的候选路径与其对应的可靠性度量值都被存储在一个列表中。该列表中所有的候选路径都被同步扩展;所以每次扩展后,列表中候选路径数量就会翻倍。此时要丢弃那些可靠性度量值较小的部分候选路径,以保证候选路径数始终不大于列表数量的一半。在译码结束时,再从该列表中找出可靠性度量值最大的那条路径,其对应的比特估计序列即为译码结果。

[0011] 串行抵消译码的另一种改良方法是串行抵消堆栈译码:使用一个有序堆栈,而不是列表,用于存储候选路径。在串行抵消堆栈译码过程中,每次都只基于可靠性度量最大的那一条候选路径(位于堆栈栈顶)进行扩展。当出现栈顶路径到达码树的某一个叶子节点时,就停止译码过程,并且将该路径所对应的比特估计序列输出,作为译码结果。

[0012] 总之,现有技术的缺点是:实用编码系统的码长是不可能无限长的,而对有限个信道进行极化操作后,仍然会存在一部分传输性能不是特别好、也不是特别差的信道。因此,有限码长的极化码在现有的译码方法下所得到的抗噪性能并不理想。而现有译码方法都是独立于循环冗余校验而进行的,没有使用“发送序列的循环冗余校验值为零”这个有利信息。因此,如何对这种译码方法进行改进,就成为业内科技人员关注的新课题。

## 发明内容

[0013] 本发明的目的是提供一种循环冗余校验辅助的极化码译码方法,以便能够大大提高采用极化码作为信道编码的通信系统的纠错能力,且其操作比较简单,运算复杂度与WCDMA系统中使用Turbo码的编译码方法相当、甚至更低,因此该方法具有很好的实用化前景。

[0014] 为了达到上述发明目的,本发明提供了一种循环冗余校验辅助的极化码译码方法,其特征在于:对极化码进行译码时,在极化码所对应的码树上,从根节点到叶子节点、且其所对应比特估计序列的循环冗余校验值为零的所有路径中,采用列表或堆栈作为路径搜索的辅助,寻找其中可靠性度量值最大的一条路径,并输出该路径对应的比特估计序列作为译码结果;所述方法包括以下操作步骤:

[0015] (1) 根据搜索辅助方法确定参数:对码长为N的极化码进行译码,设定搜索宽度为L,并确定使用列表还是堆栈进行路径搜索;若使用堆栈译码,还需设定堆栈最大深度D;其中,N、L、D都是自然数;

[0016] (2) 构造译码方法的辅助结构,包括下述两种操作内容:

[0017] 若选用列表译码,则构造一个包含 $2 \times L$ 条路径记录的列表,该列表中的每条路径记录分别用于存储一个最大长度为N的比特估计序列及其对应的可靠性度量值;

[0018] 若选用堆栈译码,则构造一个最大深度为D的有序堆栈,该堆栈中的每条记录用于存储一个最大长度为N的比特估计序列及其对应的可靠性度量值;且堆栈中的记录是根据可靠性度量值的数值从大到小顺序排列,即位于堆栈顶部的记录的可靠性度量值最大;再构造用自然数对其进行编号排序的N个计数器;

[0019] (3) 搜索候选比特估计序列:使用列表或堆栈分别在码树上进行搜索,得到可靠

性度量值最大的 L 条互不相同的从根节点到叶子节点的长度为 N 的路径, 这 L 条路径所对应的比特估计序列即为 L 个候选比特估计序列;

[0020] (4) 执行循环冗余校验: 按照可靠性度量值从大到小的顺序, 对每个候选比特估计序列进行循环冗余校验, 一旦遇到某个比特估计序列的校验值为零时, 就停止译码, 并输出该比特估计序列作为译码结果; 如果对步骤 (3) 得到的 L 个候选比特估计序列进行循环冗余校验的过程中没有校验值为零, 则译码失败。

[0021] 所述极化码的码树结构如下: 一个码长为 N 的极化码所对应的码树是一棵由  $(2^{N+1}-1)$  个节点和  $(2^{N+1}-2)$  条边组成的满二叉树, 其中除去叶子节点以外的每个节点, 与其左、右两侧两个后继节点之间的边都分别被标记为 0 和 1, 并对该码树中的边根据其到根节点所需经过的节点数进行分层, 故序号为 d 的层是由  $2^d$  条不同的边所组成; 其中, 自然数 d 的取值范围为 1 到 N, 即直接连接根节点的边的层序号  $d = 1$ , 而连接叶子节点的边的层序号  $d = N$ ; 从序号  $d = 1$  的层到序号  $d = i$  的层, 分别逐层选取一条边, 就形成一条包含  $i$  条边的路径, 再将该路径中各边的标记 0 或 1 按照层序号从小到大的顺序排列, 就能得到一个长度为  $i = i$  的比特估计序列  $(\hat{u}_1, \hat{u}_2, \dots, \hat{u}_i)$ ;

[0022] 为方便表示, 使用符号  $x_1^N$  表示实数序列或二进制的比特序列  $(x_1, x_2, \dots, x_N)$ , 再用符号  $x_s^t$  表示  $x_1^N$  的一个子序列  $(x_s, x_{s+1}, \dots, x_{t-1}, x_t)$ , 即用下标 s 和上标 t 分别表示子序列中第一个元素和最后一个元素在其原序列中的序号, 其中  $1 \leq s \leq N, 1 \leq t \leq N$ ; 且当  $s > t$  时,  $x_s^t$  表示一个空序列, 记作  $\phi$ , 空序列的长度  $l = 0$ ; 还设置  $x_1^N$  的子序列  $x_s^t$  中分别由奇数序号的元素所构成的奇数子序列为  $x_{s,o}^t$ , 以及由偶数序号的元素所构成的偶数子序列为  $x_{s,e}^t$ 。

[0023] 当设定固定比特序列为一个全零序列时, 对于给定的接收信号  $y_1^N$ , 其长度为  $i = i$  的比特估计序列  $\hat{u}_1^i$  的可靠性度量值  $R(\hat{u}_1^i | y_1^N)$  的计算方法有下述三种:

[0024] 第 1 种: 比特估计序列为空序列  $\phi$  时, 其可靠性度量值  $R(\phi | y_1^N) = 0$ ;

[0025] 第 2 种: 若序号为  $i$  的比特属于信息比特时, 其可靠性度量值的计算公式为:

$$R(\hat{u}_1^i | y_1^N) = M_N^{(i)}(\hat{u}_1^i | y_1^N) + f(\hat{u}_1^i); \text{ 式中, } M_N^{(i)}(\hat{u}_1^i | y_1^N) = \ln \left( \frac{W_N^{(i)}(y_1^N, u_1^{i-1} = \hat{u}_1^{i-1} | u_i = \hat{u}_i)}{\sum_{u_1^{i-1} \in \mathcal{X}^{i-1}} W_N^{(i)}(y_1^N, u_1^{i-1} = v_1^{i-1} | u_i = v_i)} \right),$$

集合  $\mathcal{X} = \{0, 1\}$ ,  $\mathcal{X}^k$  表示 k 个  $\mathcal{X}$  作笛卡尔积所得到的集合,  $W_N^{(i)}(y_1^N, u_1^{i-1} | u_i)$  为序号为  $i$  的比特所对应的极化信道的转移概率函数, 即发送信号  $u_i$  通过极化信道  $W_N^{(i)}$  得到输出  $y_1^N$  和  $u_1^{i-1}$  的概率,  $\ln(\cdot)$  表示以自然对数 e 为底的对数函数; 函数  $f(\hat{u}_1^i)$  的取值规则是: 当序列  $\hat{u}_1^i$  中属于固定比特的部分取值都为零时,  $f(\hat{u}_1^i) = 0$ , 否则,  $f(\hat{u}_1^i) = -\infty$ ;  $\infty$  为无穷大的实数;

[0026]  $W_N^{(i)}(y_1^N, u_1^{i-1} | u_i)$  为序号为  $i$  的比特所对应的极化信道的转移概率函数,

[0027] 第 3 种: 若序号为  $i$  的比特属于固定比特时, 其可靠性度量值的计算公式为:

$$R(\hat{u}_1^{i-1}, \hat{u}_i = 0 | y_1^N) = R(\hat{u}_1^{i-1} | y_1^N), \quad R(\hat{u}_1^{i-1}, \hat{u}_i = 1 | y_1^N) = -\infty.$$

[0028] 所述可靠性度量值中的  $M_N^{(i)}(\hat{u}_1^i | y_1^N)$  的计算方法是利用信道极化的递归结构并按照下述递归运算进行的：

[0029] 当  $i$  为奇数时,令  $i = 2k-1$ ,

[0030]

$$\begin{aligned} M_N^{(i)}(u_1^i = \hat{u}_1^i | y_1^N) &= M_N^{(2k-1)}(u_1^{2k-1} = \hat{u}_1^{2k-1} | y_1^N) \\ &= \max^* \left\{ \left[ M_{N/2}^{(k)}(u_k = \hat{u}_{2k-1}, u_1^{k-1} = \hat{u}_{1,o}^{2k-2} \oplus \hat{u}_{1,e}^{2k-2} | y_1^{N/2}) + M_{N/2}^{(k)}(u_k = 0, u_1^{k-1} = \hat{u}_{1,e}^{2k-2} | y_{N/2+1}^N) \right], \right. \\ &\quad \left. \left[ M_{N/2}^{(k)}(u_k = \hat{u}_{2k-1} \oplus 1, u_1^{k-1} = \hat{u}_{1,o}^{2k-2} \oplus \hat{u}_{1,e}^{2k-2} | y_1^{N/2}) + M_{N/2}^{(k)}(u_k = 1, u_1^{k-1} = \hat{u}_{1,e}^{2k-2} | y_{N/2+1}^N) \right] \right\} \end{aligned}$$

[0031] 当  $i$  为偶数时,令  $i = 2k$ ,

[0032]

$$M_N^{(i)}(u_1^i = \hat{u}_1^i | y_1^N) = M_N^{(2k)}(u_1^{2k} = \hat{u}_1^{2k} | y_1^N) = M_{N/2}^{(k)}(u_1^k = \hat{u}_{1,o}^{2k} \oplus \hat{u}_{1,e}^{2k} | y_1^{N/2}) + M_{N/2}^{(k)}(u_1^k = \hat{u}_{1,e}^{2k} | y_{N/2+1}^N);$$

[0033] 其中,运算符号  $\oplus$  表示模二加操作,  $u_1^N \oplus v_1^N$  表示对这两个序列逐比特做模二加操作,即  $u_1^N \oplus v_1^N = (u_1 \oplus v_1, u_2 \oplus v_2, \dots, u_N \oplus v_N)$ ;  $\max^*(a, b)$  表示雅可比对数式,即  $\max^*(a, b) = \ln(e^a + e^b) = \max(a, b) + \ln(1 + e^{-|a-b|})$ ,  $\max(a, b)$  表示选取  $a, b$  中的较大值,式中  $a$  和  $b$  均为实数;因计算  $\max^*(a, b)$  需要的计算资源多,且有对数和指数运算,硬件不易实现;故预先为函数  $g(c) = \ln(1 + e^{-c})$  设置一个表,其定义域为大于或等于零的实数,即  $c \geq 0$ ;这样计算  $\max^*(a, b) = \max(a, b) + g(|a-b|)$  时,其中的  $g(|a-b|)$  就直接查表;且为继续降低复杂度,能直接用  $\max(a, b)$  代替  $\max^*(a, b)$  操作,但这样会引入一定程度的性能损失。

[0034] 所述步骤(3)使用列表的候选比特估计序列进行搜索译码时,首先执行下述初始化操作:

[0035] 清空列表后,向该列表中添加一条其内容为一个长度为零的空序列  $\Phi$  的记录,且设置其可靠性度量值为零。

[0036] 所述步骤(3)使用列表的候选比特估计序列进行搜索译码时,在初始化操作后,执行下列操作内容:

[0037] (3A) 扩展列表中的候选比特估计序列:此时列表中各个记录的比特估计序列具有相等的长度1,逐一取出列表中的各条记录,并在被取出的每条记录中的比特估计序列后分别添加一个0比特或一个1比特,得到两个新的长度为1+1的比特估计序列;接着分别计算这两个长度为1+1的比特估计序列的可靠性度量值,并以该两个新的比特估计序列及其对应的可靠性度量值为内容生成两条记录;待列表中所有记录被全部取出后,将新产生的记录存入列表中;

[0038] (3B) 列表中候选比特估计序列进行竞争:完成步骤(3A)操作后,如果列表中的记录数量小于或等于L,则直接跳过本步骤而执行步骤(3C);如果列表中的记录数量大于L,则保留可靠性度量值最大的L项,并删除其余各项;

[0039] (3C) 判断是否停止搜索:判断此时列表中记录中的比特估计序列长度是否小于

N,若是,则返回继续重新执行步骤(3A);否则,此时列表中各个记录中的比特估计序列即为步骤(4)所需要的候选比特估计序列,结束步骤(3)的操作,执行后续步骤(4)。

[0040] 所述步骤(3)使用堆栈的候选比特估计序列进行搜索译码时,首先执行下述初始化操作:

[0041] 用非负整数序列 $q_1^N = (q_1, q_2, \dots, q_N)$ 中的各个元素分别表示各计数器的数值,并设其为一个全零序列,在清空堆栈后;向堆栈中添加一条其内容为一个长度为零的空序列 $\Phi$ 的记录,且设置其可靠性度量值为零。

[0042] 所述步骤(3)使用堆栈的候选比特估计序列进行搜索译码时,在初始化操作后,执行下列操作内容:

[0043] (3a) 执行弹栈操作:将堆栈顶部的记录弹出,查得该记录的所对应的比特估计序列长度为1,若1不为0,就将序号为1的计数器的数值 $q_1$ 加1;

[0044] (3b) 栈中候选比特估计序列进行竞争:如果序号为1的计数器的数值已经达到了L,即 $q_1 = L$ ,则检查堆栈中的所有记录,并删除所有比特估计序列长度小于或等于1的记录;然后,判断堆栈最大深度减去当前堆栈中的记录数量之差是否大于或等于2,如果是,则删除当前位于堆栈底部的那条记录;否则,执行后续步骤(3c);

[0045] (3c) 扩展堆栈中的候选比特估计序列:在步骤(3a)中被弹出记录中的比特估计序列后添加一个0比特或一个1比特,得到两个新的长度为1+1的比特估计序列,接着分别计算这两个长度为1+1的比特估计序列的可靠性度量值;

[0046] (3d) 执行压栈操作:以步骤(3c)中得到的长度为1+1的两个新比特估计序列及其各自的可靠性度量值为内容,生成两条新记录,并压入堆栈中;再按照可靠性度量值从大到小的顺序重新排列堆栈中的记录,以使位于堆栈顶部的记录具有最大可靠性度量值;

[0047] (3e) 判断是否停止搜索:判断此时堆栈中是否已经没有任何记录,若是,则结束该步骤(3)的操作,执行后续步骤(4);否则,再判断位于堆栈顶部的记录中的比特估计序列的长度是否为N,若是,则将该记录弹出堆栈,该记录中存储的比特估计序列作为步骤(4)所需的一个候选比特估计序列,并使序号为N的计数器的数值 $q_N$ 加1后,接着判断该序号为N的计数器数值 $q_N$ 是否已经等于或大于L,若是,则结束该步骤(3)的操作,执行后续步骤(4);否则,返回执行步骤(3a)。

[0048] 所述步骤(3)中,采用列表译码时,L个候选比特估计序列是同时得到的;而采用堆栈译码时,L个候选比特估计序列是按照可靠性度量值从大到小的顺序逐一分别得到的。

[0049] 采用堆栈译码方法时,能够将路径搜索步骤中的(3e)和步骤(4)的循环冗余校验结合起来,使得路径搜索过程能够提前结束,以降低计算复杂度;此时,执行的操作内容是循环冗余校验辅助下比特估计序列的选择:若堆栈中已经没有任何记录,就停止译码,表明译码失败;如果位于堆栈顶部的记录中的比特估计序列的长度为N,则将该记录弹出堆栈,并使序号为N的计数器的数值 $q_N$ 加1;然后,对该比特估计序列进行循环冗余校验;若校验值为零,则停止译码并输出该比特估计序列作为译码结果;若校验值不为零,且 $q_N$ 已等于或大于L,则停止译码,表明译码失败;否则,返回执行步骤(3a)。

[0050] 本发明循环冗余校验辅助的极化码译码方法的技术关键是:首先定义比特估计序列的可靠性度量值,并设计该可靠性度量值的迭代计算方法。另外,提供了实用的循环冗余校验辅助的列表译码和堆栈译码的两种方法,其中,列表译码方法是在列表译码过程停止

时,按照可靠性度量值的从大到小顺序,逐一对列表中所有候选比特估计序列进行循环冗余校验,并输出第一个校验值为零的序列作为译码结果;而不是具有最大可靠性度量值的那个序列。堆栈译码方法是引入搜索宽度,允许译码过程中多次输出长度等于码长的比特候选序列,而不是如同传统的堆栈译码方法,只要遇到一条就停止搜索。每当译码方法输出一条长度等于码长的比特候选序列,就对其进行循环冗余校验,若校验值为零,则将其最终输出作为译码结果。

[0051] 因此,与现有技术相比较,本发明译码方法的优点是:利用了送入极化码编码器比特序列的循环冗余校验值为零的信息,在没有付出额外计算复杂度的前提下,相比已有的极化码译码方案,具有更强的纠错能力。另外,本发明在对有限码长的极化码进行译码时,能获得与 WCDMA 系统中使用 Turbo 码的编译码方案更好的性能,并且其运算复杂度与其相当、甚至更低。由于具备上述特性,使得本发明方法更适合应用于实际通信系统,因此,其推广应用前景看好。

## 附图说明

[0052] 图 1 是使用循环冗余校验检错的编码通信方案操作步骤流程方框图。

[0053] 图 2 是信道极化的基本单元结构组成示意图。

[0054] 图 3 是长度为 N 的信道极化装置的递归结构组成示意图,其中递归的最小单元(即  $N = 1$  时)为图 2 所示的基本单元。

[0055] 图 4 是一个码长  $N = 4$  的极化码的码树,图中黑实线指示了一条串行抵消译码得到的路径,对应的比特估计序列为 (0110)。

[0056] 图 5 是本发明循环冗余校验辅助的极化码译码方法操作步骤流程图。

[0057] 图 6 是本发明极化码译码方法采用列表的码树扩展示意图。

[0058] 图 7 是本发明循环冗余校验辅助的列表译码方法操作步骤流程图。

[0059] 图 8 是本发明极化码译码方法采用堆栈的码树扩展示意图。

[0060] 图 9 是本发明循环冗余校验辅助的堆栈译码方法操作步骤流程图。

[0061] 图 10 是本发明循环冗余校验辅助的列表译码性能曲线图。

[0062] 图 11 是本发明循环冗余校验辅助的堆栈译码性能曲线图。

[0063] 图 12 是各种译码方法的平均计算复杂度曲线。

## 具体实施方式

[0064] 为使本发明的目的、技术方案和优点更加清楚,下面结合附图和仿真实施例对本发明的实施过程及性能分析作进一步的详细描述。

[0065] 参见图 5,介绍本发明循环冗余校验辅助的极化码译码方法:在对极化码进行译码时,在极化码所对应的码树上,从根节点到叶子节点、且其所对应比特估计序列的循环冗余校验值为零的所有路径中,采用列表或堆栈作为路径搜索的辅助,寻找其中可靠性度量值最大的一条路径,并输出该路径对应的比特估计序列为译码结果。该方法包括以下四个操作步骤:

[0066] 步骤 1,根据搜索辅助方法确定参数:对码长为 N 的极化码进行译码,设定搜索宽度为 L,并确定使用列表还是堆栈进行路径搜索;若使用堆栈译码,还需设定堆栈最大深度

D ;其中, N、L、D 都是自然数 ;

[0067] 步骤 2,构造译码方法的辅助结构,包括下述两种操作内容 :

[0068] 若选用列表译码,则构造一个包含  $2 \times L$  条路径记录的列表,该列表中的每条路径记录分别用于存储一个最大长度为 N 的比特估计序列及其对应的可靠性度量值 ;

[0069] 若选用堆栈译码,则构造一个最大深度为 D 的有序堆栈,该堆栈中的每条记录用于存储一个最大长度为 N 的比特估计序列及其对应的可靠性度量值 ;且堆栈中的记录是根据可靠性度量值的数值从大到小顺序排列,即位于堆栈顶部的记录的可靠性度量值最大 ;再构造用自然数对其进行编号排序的 N 个计数器 ;

[0070] 步骤 3,搜索候选比特估计序列 :使用列表或堆栈分别在码树上进行搜索,得到可靠性度量值最大的 L 条互不相同的从根节点到叶子节点的长度为 N 的路径,这 L 条路径所对应的比特估计序列即为 L 个候选比特估计序列 ;

[0071] 步骤 4,执行循环冗余校验 :按照可靠性度量值从大到小的顺序,对每个候选比特估计序列进行循环冗余校验,一旦遇到某个比特估计序列的校验值为零时,就停止译码,并输出该比特估计序列作为译码结果 ;如果对步骤 (3) 得到的 L 个候选比特估计序列进行循环冗余校验的过程中没有校验值为零,则译码失败。

[0072] 本发明已经进行了多次实施试验,下面简要说明实施例的情况 :

[0073] 循环冗余校验码的生成多项式为  $g(D) = D^{24} + D^{23} + D^6 + D^5 + D + 1$ , 其校验比特长度  $k = 24$ , 极化码的码长  $N = 1024$ , 码率  $R = 0.5$ 。再以 WCDMA 系统中的 Turbo 码作为实施例的对比方案。Turbo 码的分量码为两个 8 状态递归系统卷积码,其生成多项式为  $[1, 1+D+D^3/1+D^2+D^3]$ , 并使用速率比配算法调整码率为  $R = 0.5$ 、码长  $N = 1024$ 。两者的信道一致,都是二进制相移键控 (BPSK) 调制的加性高斯白噪声 (AWGN) 信道。

[0074] 先参见图 6 和图 7,介绍采用循环冗余校验辅助的列表译码方法 :

[0075] 设定译码方法的搜索宽度  $L = 32$ ,对接收到的信号按照图 7 所示的本发明循环冗余校验辅助的列表译码方法的操作流程进行译码,其误码块性能如图 10 所示。作为参照,图 10 中还给出了原有的串行抵消译码和串行抵消列表译码两种方法的性能曲线以及 Turbo 码在使用对数最大后验概率迭代译码算法 (最大迭代次数设定为 8 次) 下的译码性能曲线。

[0076] 从四条性能曲线可以看出,极化码在使用本发明循环冗余校验辅助的列表译码时,其性能明显优于两种对比的已有极化码译码方法。与使用对数最大后验概率译码的 Turbo 方法相比较,极化码使用循环冗余校验辅助的列表译码方法还能够获得可观的 0.5dB 增益。

[0077] 再参见图 8 和图 9,介绍采用循环冗余校验辅助的堆栈译码方法 :

[0078] 设定译码方法的搜索宽度  $L = 32$ ,最大堆栈深度  $D = 1000$ ,对接收到的信号按照图 9 所示的循环冗余校验辅助的堆栈译码方法的操作流程进行译码,其误码块性能如图 11 所示。作为参照,图 11 中还给出了原有的串行抵消译码和串行抵消堆栈译码两种方法的性能曲线以及 Turbo 码在使用对数最大后验概率迭代译码算法 (最大迭代次数设定为 8 次) 下的译码性能曲线。

[0079] 从四种不同的性能曲线可以得到与循环冗余校验辅助的列表译码类似的结论。并且,结合图 10 与图 11,可以发现在搜索宽度设置相同时,两种译码方法获得的性能大致是

一样的。

[0080] 接着,对本发明循环冗余校验辅助的极化码译码的两种方法进行复杂度分析(参见图12):通过对一个码块完成译码的过程中度量值更新操作的平均次数的统计数,来衡量各种极化码译码算法的复杂度。等价地,对Turbo码是统计对数最大后验概率译码过程中四类度量值的平均更新次数。图12是这些译码方法的平均复杂度的统计曲线。

[0081] Turbo码在进行译码时,每经过一次迭代都要进行一次循环冗余校验,如果校验值为零,则直接停止译码操作,因此其复杂度是随着信噪比的提高而降低的。对于极化码的循环冗余校验辅助列表译码,由于其搜索范围是固定的,不会随着信噪比的改变而明显变化(其发生轻微变化的原因是极化码的构造会随着信噪比而改变)。而循环冗余校验辅助堆栈译码方法则不同,随着信噪比的提高,正确的路径会越来越容易地保持其位于栈顶的位置,从而计算复杂度会明显下降。可以看到,循环冗余校验辅助的堆栈译码具有与Turbo码相当的计算复杂度。虽然循环冗余校验辅助的列表译码比循环冗余校验辅助堆栈译码的计算复杂度更高些,但是前者对存储空间的开销明显较后者要小。

[0082] 以上所述仅为本发明的一个实施案例的介绍而已,并不用以限制本发明,凡在本发明的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本发明保护的范围之内。

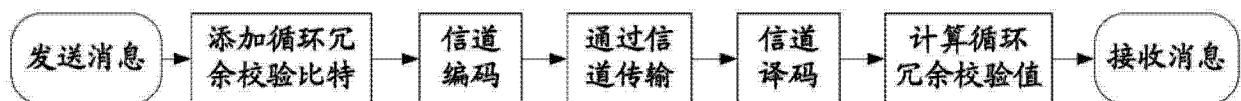


图 1

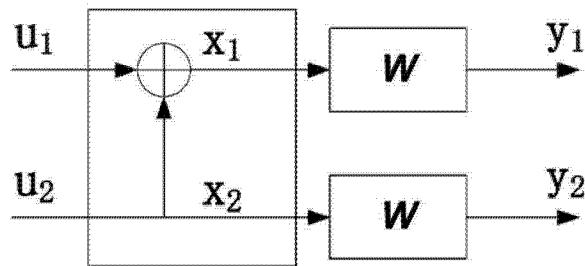


图 2

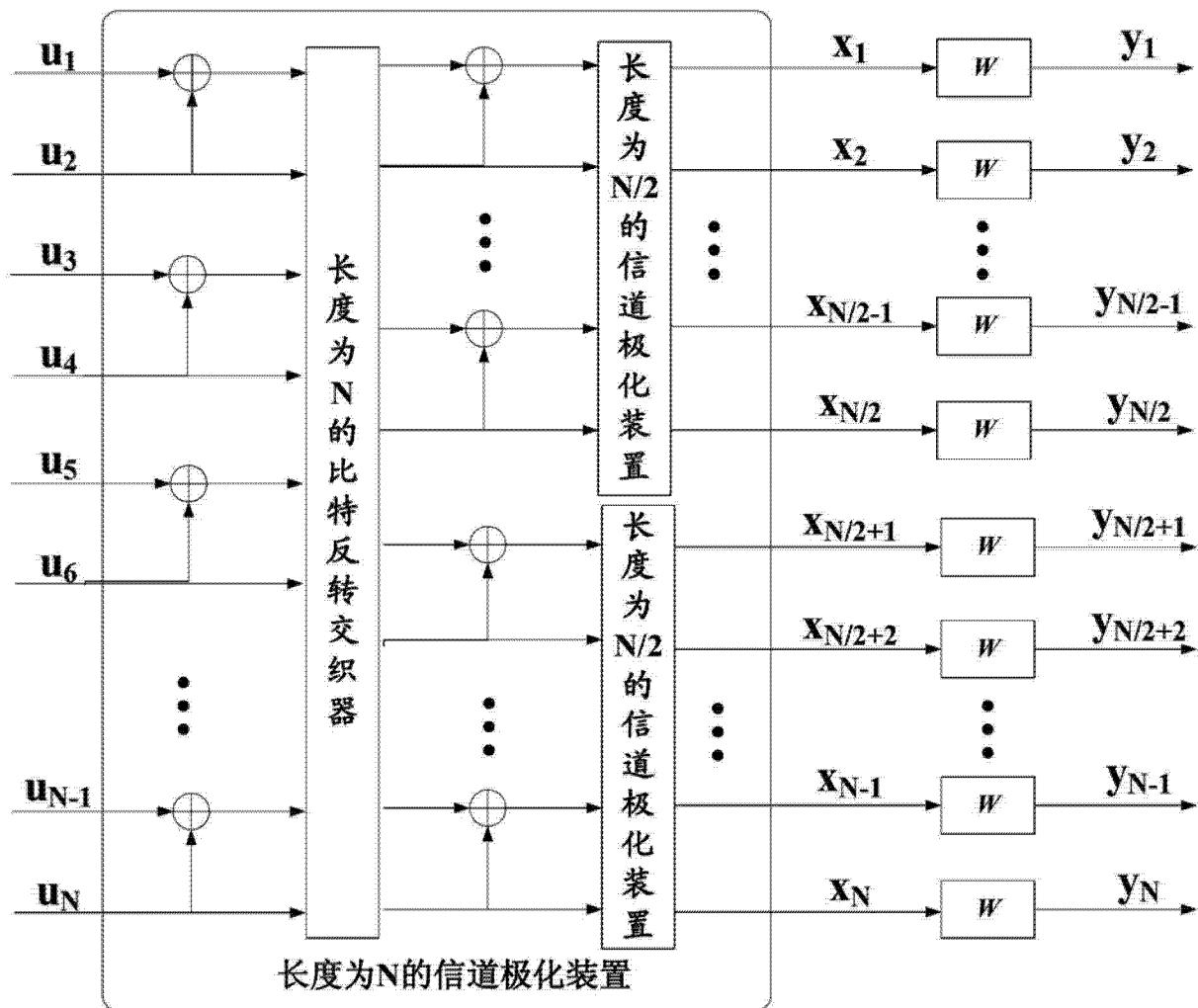


图 3

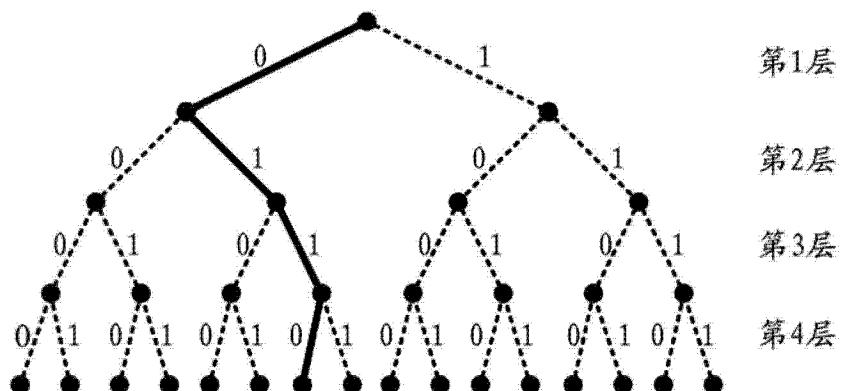


图 4

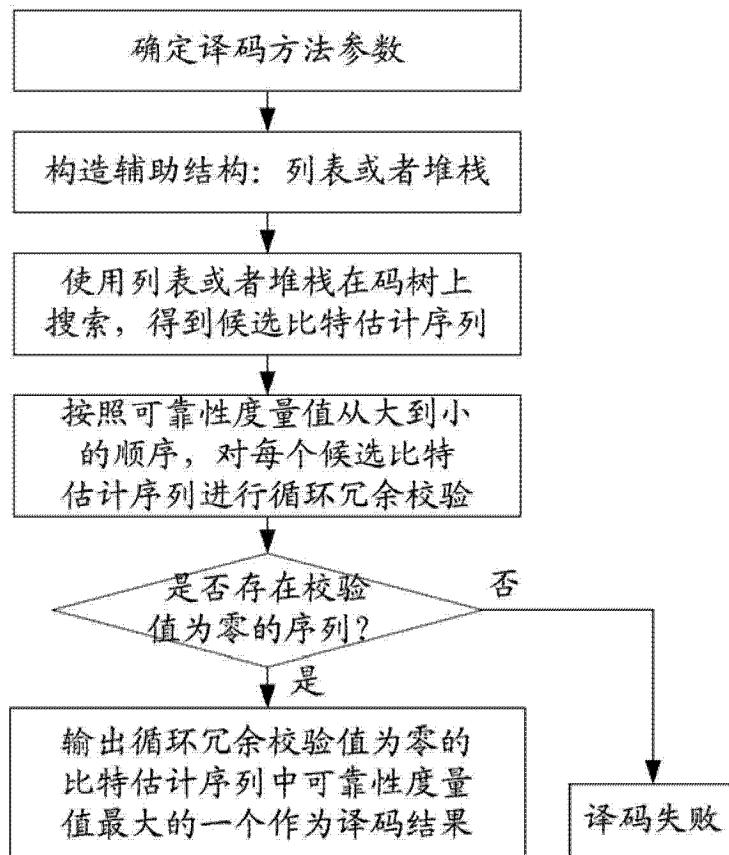


图 5

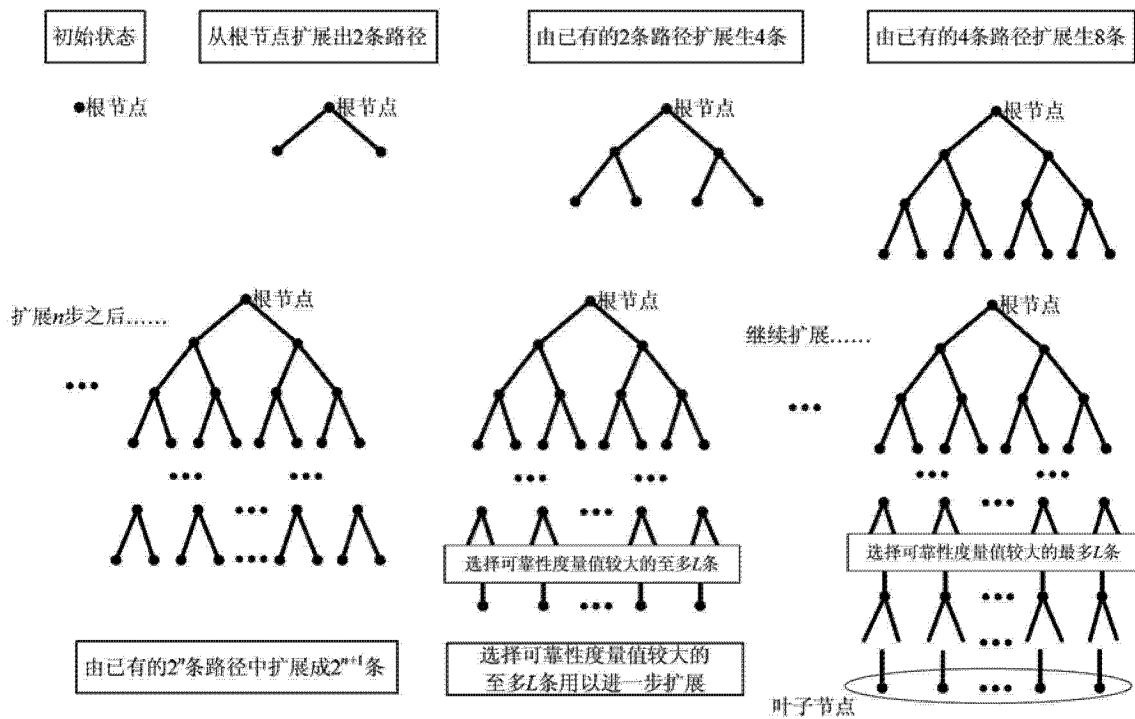


图 6

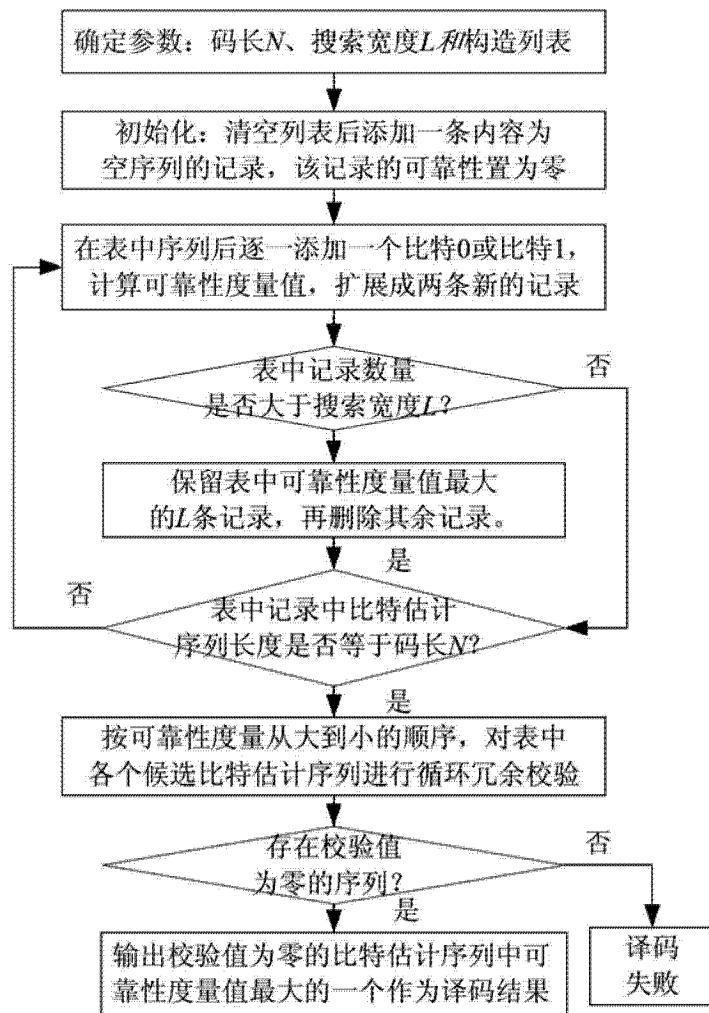


图 7

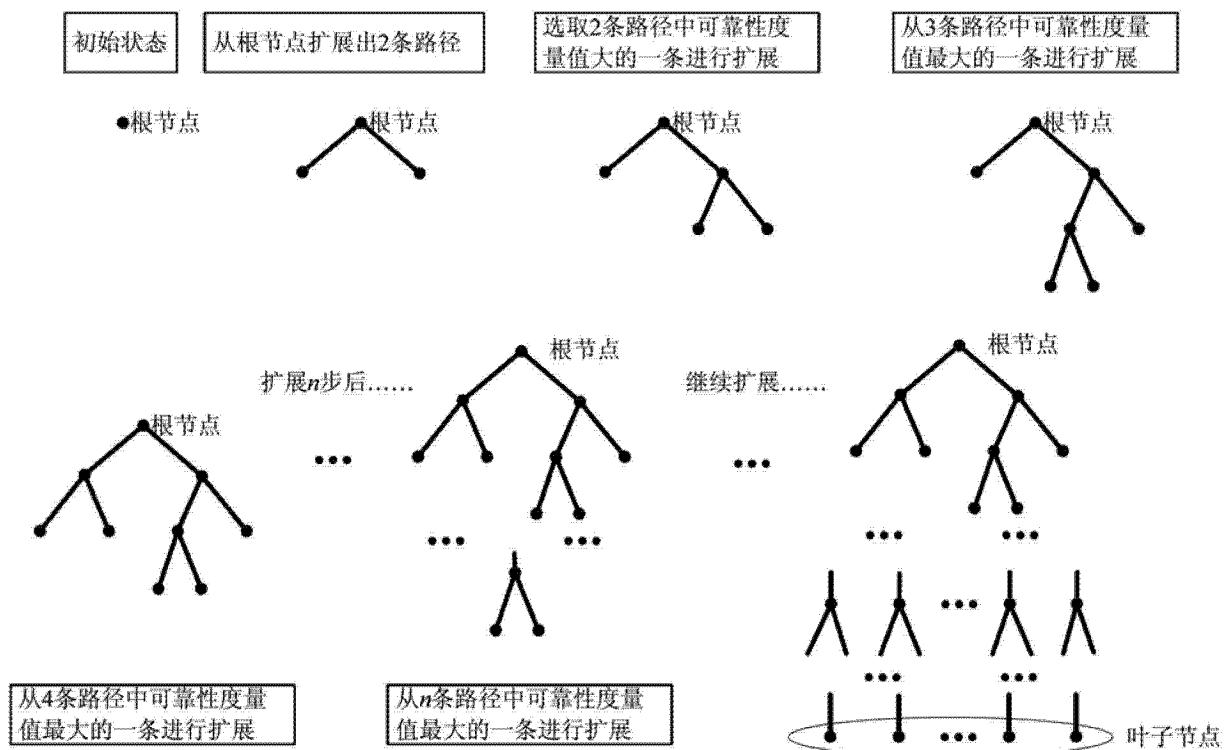


图 8

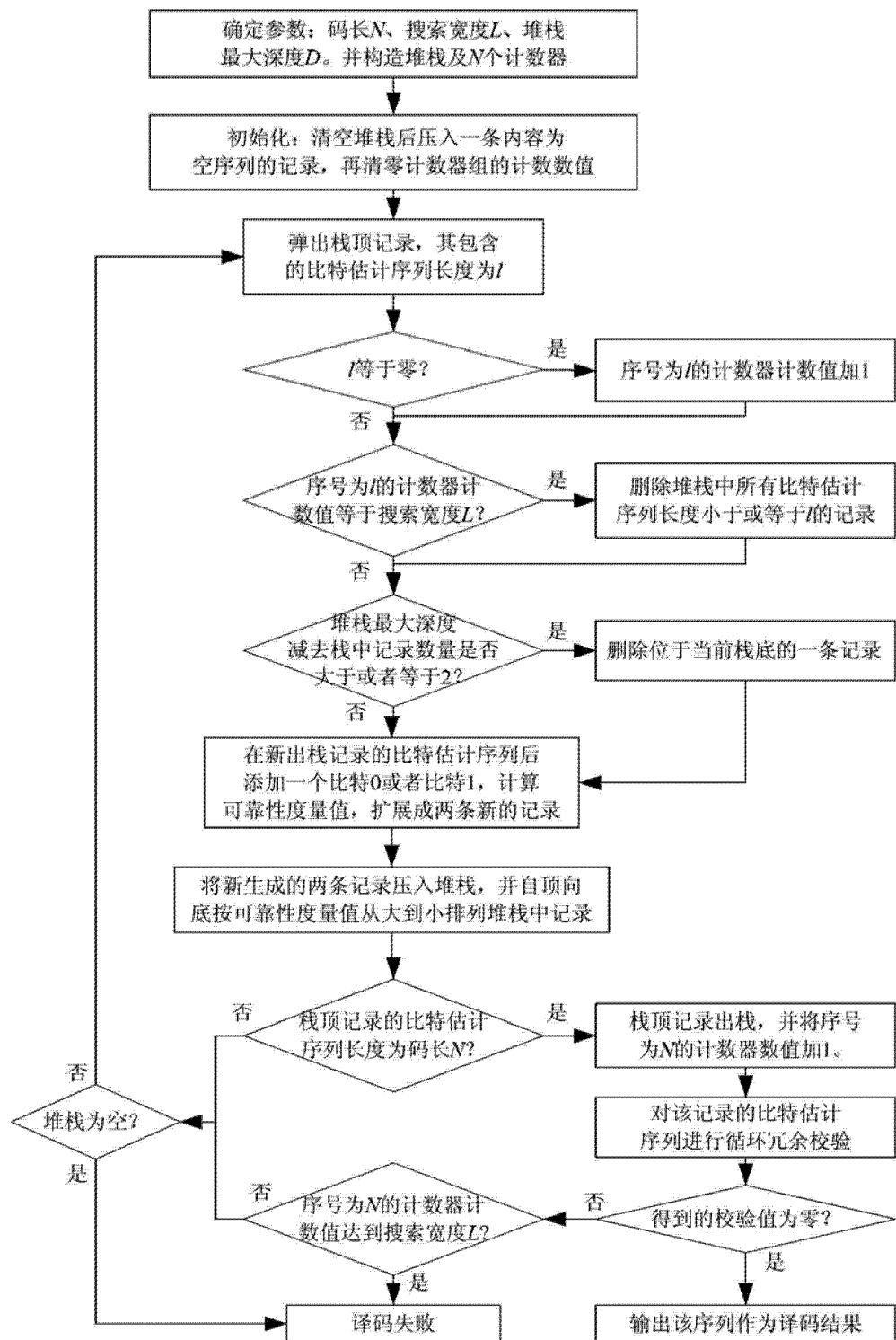


图 9

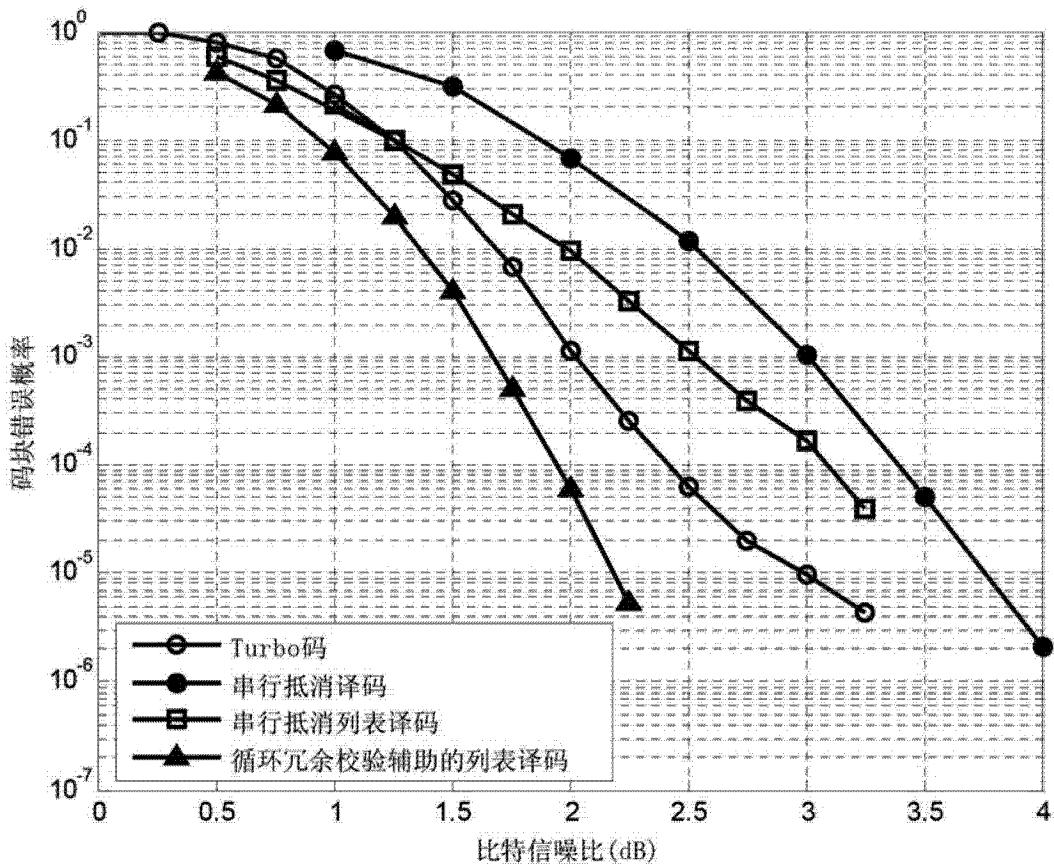


图 10

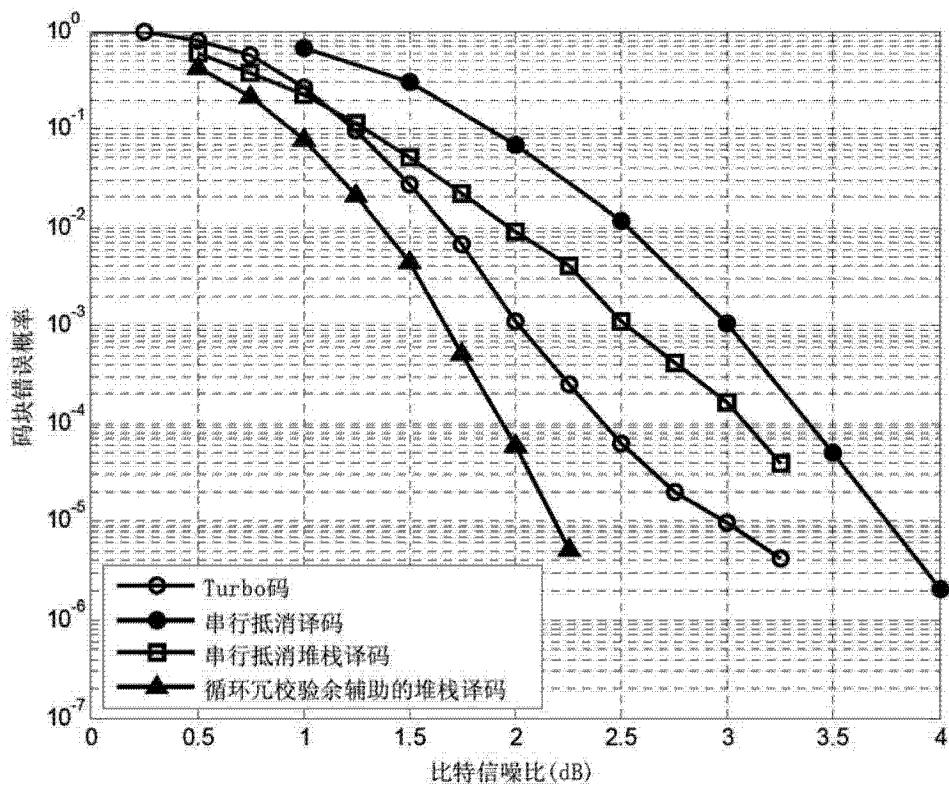


图 11

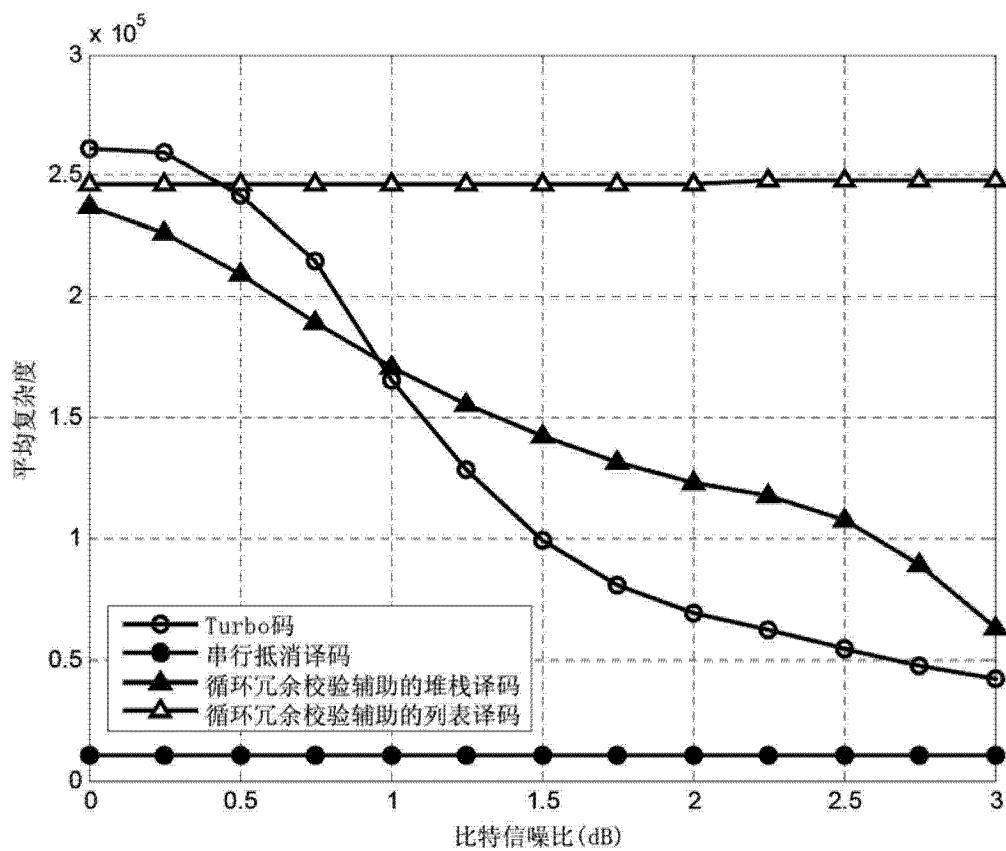


图 12