文章编号:1001 - 9081(2010)07 - 1760 - 03

TPC 基于相关运算的迭代译码算法

王 玮,葛临东,巩克现
(信息工程大学 信息工程学院,郑州 450002)
(wangwei1984318@yahoo.com.cn)

摘 要:Chase-Pyndiah 算法(简称 C-P 算法)为 Turbo 乘积码(TPC)译码中常采用的算法之一。在 C-P 算法的基础上,引入一种基于相关运算的迭代译码算法,采用相关作为度量,可以避免复杂的欧氏距离计算;在选择候选码字时引入度量比较的方法,省去了对竞争码字的搜索;通过去除候选码字中相同元素对符号集合进行简化,降低了译码复杂度和译码延时。经算法分析与仿真表明,与已有的软判决算法相比,该算法的译码速度更快而译码性能没有降低,非常适合硬件实现。

关键词:Turbo 乘积码;Chase-Pyndiah(C-P)算法;迭代译码;复杂度;延时
中图分类号:TP393 **文献标志码**:A

Iterative decoding algorithm of TPC based on correlation computation

WANG Wei, GE Lin-dong, GONG Ke-xian
(*College of Information Engineering, Information Engineering University, Zhengzhou Henan* 450002, *China*)

Abstract: Chase-Pyndiah algorithm is one of the decoding algorithms of Turbo Product Code (TPC). Based on that, an iterative decoding algorithm involved with correlation was introduced. This algorithm avoids the computation of Euclidean distance due to the use of correlation as the metric; and uses a comparative method in the choice of the concurrent codewords, which prevents the search of competition codeword. On the basis of this study, further simplification was presented, which decreased the complexity and the latency of the decoder. Both analysis and simulation result show the advantage of the proposed algorithm over conventional algorithm in speed and performance.

Key words: Turbo Product Code (TPC); Chase-Pyndiah (C-P) alogrithm; iterative decoding; complexity; latency

0 引言

1994 年,Pyndiah 等人[1]在 Chase 算法的基础上提出了一种线性分组码的软输入软输出迭代译码算法,并将它应用于乘积码译码中,获得了很好的编码增益。由于其性能与 Turbo 卷积码较为相近,1998 年,Comtech AHA 公司将采用这种译码方式的乘积码称为 Turbo 乘积码(Turbo Product Code, TPC)。基于 Chase 算法的 TPC 迭代译码算法是一种通过缩小码字搜索范围的次最优译码算法。在译码时,将码元的对数似然比(Log-Likelihood Ratio, LLR)作为译码器的软输出,并将软输出信息减去软输入信息,作为下次迭代的外信息。通过这种方式不断修正码元的软信息,增大其可靠度,获得较好的译码性能。由于 Turbo 乘积码的译码性能十分接近香农限,并且与 Turbo 卷积码(Turbo Convolutional Code, TCC)相比,不仅在高码率时能够很好地避免"地板效应",同时译码复杂度大大降低,在通信领域引起了广泛的关注。在 Chase 算法中,译码时码字的搜索范围由 2^k 降低为 2^p [2](k 为信息位个数,p 为不可靠位个数)。但是,由于计算最佳码字时采用欧氏距离作为度量,其译码复杂度始终得不到有效降低,并且在搜索竞争码字的同时,每一位外信息的计算都需要进行欧氏距离的比较,由此带来了较大的复杂度以及译码延迟。如何在保证性能的前提下,有效简化 Chase 算法,解决其应用瓶颈,成为一个突出的关键问题。近年来,也有不少学者提出对 TPC 译码算法的改进,如梯度算法[2]、并行译码算法[4]等。虽然这些算法都在一定程度上减少了运算量,降低了系统延迟,但由于计算中采用了某些近似,不可避免地带来一定的性能损失。本文在 Chase 算法的基础上,在不降低译码性能的前提下,采用相关作为码字度量,避免了欧氏距离的复杂计算,省去了对竞争码字的重复搜索;并在此基础上做了进一步改进,大大减少了计算量,降低了译码延时。

1 TPC 编码方法

乘积码的概念最早由 Elias 于 1954 年提出[5],它将多个短码通过行列交织方式级联构成二维或多维码字,这是一种增大码字最小汉明距离的简单而有效的方法,首次实现了非零码率的无误码率传输。从码字分类上讲,乘积码实际上是一种串行级联码,且子码级联顺序可以任意交换而不影响码字整体的结构与性能,具有非常灵活的编译码特点。以二维乘积码为例,其编码过程如下:

1)将信息位排列成 $k_1 \times k_2$ 的信息矩阵 $\boldsymbol{C}$;

2)用 $C^1(n_1,k_1,d_1)$ 码对 $\boldsymbol{C}$ 的每一行进行编码,得到 $C_1(k_2 \times n_1)$;

3)用 $C^2(n_2,k_2,d_2)$ 码对 C_1 的每一列进行编码,得到乘积码字 $E(n_1 \times n_2)$。

所得乘积码的码长为 $n = n_1 \times n_2$,信息位为 $k = k_1 \times k_2$,码字间最小距离为 $d = d_1 \times d_2$。编码结构如图 1 所示。

收稿日期:2010 - 01 - 14;**修回日期**:2010 - 03 - 04。 **基金项目**:河南省基础与前沿项目(082300413205)。

作者简介:王玮(1984 -),女,河北衡水人,硕士研究生,主要研究方向:软件无线电、信道编译码; 葛临东(1946 -),男,安徽怀远人,教授,主要研究方向:软件无线电、信号与信息处理; 巩克现(1976 -),男,山东新泰人,副教授,主要研究方向:软件无线电、差错控制编码、联合编码调制。

在实际应用中,通常采用行列编码方式相同的码字作为子码构造 TPC 码,并且在行列编码时增加一个奇偶校验位以增大码间最小距离。子码常采用的码字有扩展汉明码、扩展 BCH(eBCH)码、RS 码等。

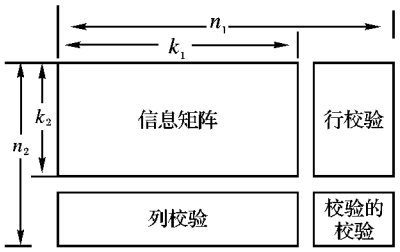

图 1　二维乘积码的编码结构

2　Chase-Pyndiah 算法

根据 TPC 的构造方法,最初采用的方法为对接收到的码字进行硬判决,在此基础上分别进行行列代数译码。但这种译码方法仅能纠正行、列有单个错误的码字,具有无法消除的"#"形译码硬伤。而采用迭代处理的软信息译码可以有效地解决硬判决译码中遇到的问题,并且获得超过 3 dB 的编码增益,成为近年来常用的一种译码算法。其中,Chase-Pyndiah 算法就是一种典型的 TPC 译码算法。

2.1　Chase 算法

1972 年,Chase 根据译码中使码字错误概率最小的原则提出一种取得近似最大似然(Maximum Likelihood,ML)序列估计的方法[2],将对最优码字的搜索范围缩小到以硬判决码字 Y 为中心、Hamming 距离为 $\delta_i - 1$ 的圆内,减少了译码复杂度。Chase 算法的具体步骤如下。

1)对接收码字 $R = (r_1, r_2, \cdots, r_n)$ 作硬判决得到码字 $\boldsymbol{Y}$,判决规则为:$y_j = 0.5(1 + \mathrm{sgn}(r_j))$。

2)将码元可靠度定义如下:

$$A(y_j) = \ln\left(\frac{P\{e_j = +1 \mid r_j\}}{P\{e_j = -1 \mid r_j\}}\right) = \left(\frac{2}{\sigma^2}\right) r_j \tag{1}$$

经归一化后,y_j 的可靠度由 $|r_j|$ 得出,以此作为码元可靠度的度量,确定 Y 的 $p = \lfloor d/2 \rfloor$ 个最不可靠位(即 $|r_j|$ 最小的 p 位),并记录其位置。

3)分别在 p 个不可靠位置上取"0"和"1",构造一个由 2^p 个 n 维向量组成的测试图样 T^q。

4)构造测试序列 $Z_j = T_j \oplus Y$,对其进行代数译码后,将正确的码字 C^i 加入到候选码字集合 Ω 中。

5)将 C^i 作{0,1}到{-1,1}的映射,求出它与接收码字 R 之间的欧氏距离,取距离最小的码字作为判决码字 D。其中,欧氏距离的计算如下:

$$|R - C^i|^2 = \sum_{l=1}^{n} (r_l - c_l^i)^2;\ c_l \in \{-1, +1\} \tag{2}$$

2.2　线性分组码的软判决译码

1994 年,Pyndiah 等人在 Chase 算法的基础上提出软输入软输出的迭代译码算法,并用对数似然比(Log Likely Ratio,LLR)定义所得到的判决码字 D 的可靠度。码元的 LLR 表示如下[1]:

$$A(d_j) = \frac{1}{2\sigma^2}(|R - C^{-1(j)}|^2 - |R - C^{+1(j)}|^2) \tag{3}$$

经化简并归一化后可得:

$$r_j' = r_j + w_j, w_j = \sum_{l=0, l\neq j}^{n-1} r_l c_l^{+1(j)} p_l \tag{4}$$

其中:r_j' 替换了 $A'(d_j)$,是译码器真正的软输出;w_j 称为外信息。

由式(3)可以看出,在软判决译码输出端计算判决码字的可靠度需要两个码字 $C^{+1(j)}$ 和 $C^{-1(j)}$。其中一个码字为最佳判决码字 D,另外一个码字称为竞争码字 C,则由式(3)、(4)可得,Chase 译码器的软输出为:

$$r_j' = \frac{1}{4}(|R - D|^2 - |R - C|^2) \times d_j \tag{5}$$

若在子集合 Ω 中找不到竞争码字 C,Pyndiah 给出了一种简单而有效的方案[7],关系式如下:

$$r_j' = \beta \times d_j + r_j \tag{6}$$

3　基于相关运算的迭代译码算法

由式(5)可知,传统 Chase 算法的迭代运算需要计算测试序列与接收码字之间的欧氏距离,从中选取距离最小的码字作为最佳判决码字 D。而在寻找竞争码字 C 时,也是在与 d_j 符号不同的码字集合中选取欧氏距离最小的码字作为竞争码字。由欧氏距离带来的计算十分复杂,成为系统的主要瓶颈。针对这一问题,文献[6]基于以下理论提出了一种采用相关作为度量的算法。

式(2)中,欧氏距离表达式可展开如下:

$$|R - C^i|^2 = \sum_{l=1}^{n} r_l^2 + \sum_{l=1}^{n} (c_l^i)^2 - 2\sum_{l=1}^{n} r_l c_l^i;\ c_l^i \in \{-1, +1\} \tag{7}$$

其中 $\sum_{l=1}^{n} r_l c_l^i$ 为 R 与 C^i 的内积,也称做 R 与 C^i 的相关。由上式可得,对于一个给定的接收码字,$\sum_{l=1}^{n} r_l c_l^i$ 越大,欧氏距离越小。因此,寻找欧氏距离最小的码字等效为寻找相关最大的码字,欧氏距离的比较可以简化为相关的比较。

令 $p(c) = \sum_{l=1}^{n} r_l c_l^i$,则式(5)经化简可表示为:

$$r_j' = \frac{1}{2}[p(c^{+1(j)}) - p(c^{-1(j)})] \tag{8}$$

其中:$p(c^{+1(j)})$ 为第 j 位为 +1 的所有码字集合中码字 C^i 与 R 的最大相关值;$p(c^{-1(j)})$ 为第 j 位为 -1 的所有码字集合中码字 C^i 与 R 的最大相关值。

基于以上分析,对外信息的计算可以简化为以下几步。

1)令 $s^+ = (s_1^+, \cdots, s_j^+, \cdots, s_n^+)$,$s^- = (s_1^-, \cdots, s_j^-, \cdots, s_n^-)$,根据 C^i 每个码元的正负值将 C^i 与 R 的相关值分类,前者用来存储码元为 +1 时的相关值,后者用来存储码元为 -1 时的相关值,初始值均置为 $-\infty$。

2)设 Chase 译码器经代数译码后输出的第一个码字为 $C_1 = (c_{1,1}, \cdots, c_{1,j}, \cdots, c_{1,n})$,其与 R 的相关值为 $p(c_1)$,令集合 $J_1^+ = \{j \in \{1,2,\cdots,n\} \mid c_{1,j} = +1\}$,$J_1^- = \{j \in \{1,2,\cdots,n\} \mid c_{1,j} = -1\}$。对 s^+ 与 s^- 分别赋值,赋值规则为:若 $j \in J_1^+$,则 $s_j^+ = p(c_1)$;若 $j \in J_1^-$,则 $s_j^- = p(c_1)$。

3)对每个码字 $C_k = (c_{k,1}, \cdots, c_{k,j} \cdots, c_{k,n})$ 以及它与 R 的相关值 $p(c_k)$,令集合 $J_k^+ = \{j \in \{1,2,\cdots,n\} \mid c_{k,j} = +1\}$,$J_k^- = \{j \in \{1,2,\cdots,n\} \mid c_{1,k} = -1\}$。若 $j \in J_k^+$,当 $p(c_k) > s_j^+$ 时,$s_j^+ = p(c_k)$;若 $j \in J_k^-$,当 $p(c_k) > s_j^-$ 时,$s_j^- = p(c_k)$;否则,值不变。

4)比较完毕,可得 $r' = (r_1', \cdots, r_j', \cdots, r_n') = \frac{1}{2}(s^+ - s^-)$。其中值为 $+\infty$ 或 $-\infty$ 的码元表示该位置不存在竞争码字,其外信息的计算由 β 得出;同时,可以得出最佳判决码字 $D = (\mathrm{sign}(r_1'), \cdots, \mathrm{sign}(r_j'), \cdots, \mathrm{sign}(r_n'))$。

由上述步骤可见,复杂的欧氏距离计算经化简,仅需求出

C^i 与 R 的相关就可得到所需要的软输出，并且在计算相关时，由于 $c_l^i \in \{-1, +1\}$，运算仅由加减法组成，大大减小了运算量。

4 对相关迭代译码算法的改进

由以上分析可知，相关迭代译码主要基于对码字相关的比较。经观察可以发现，Chase 算法中，候选码字集合 Ω 中的码字仅在不可靠位和代数译码过程中经纠错的位置符号不同。基于这一点，可对算法进行进一步简化。

4.1 对相关运算的简化

已知 Ω 中的码字都来源于硬判决码字 $\boldsymbol{Y}$，它们之间含有相同的部分。因此，可对相关运算化简如下：

1）在 Chase 算法的第4）步完成后，令 $c_{i,j}' = c_{i,j} \oplus y_j$，其中 $c_{i,j}$ 为 Ω 中第 i 个码字的第 j 位，y_j 为接收序列的硬判决码字的第 j 位；

2）相关度量可以表示为 $p(c_i) = p(y_i) + \sum_{l=1}^{n} c_{i,l}' r_l$，其中 $c_{i,l}' = 1$。令 $p(c_i') = -\sum_{l=1}^{n} c_{i,l}' |r_l| (c_{i,l}' = 1)$ 作为码字相关大小的衡量，这样就进一步减少了相关的计算量。

4.2 对寄存器 S 的简化

在软输出计算中，根据是否存在竞争码字采用两种不同的公式，因此可将不存在竞争码字的位置剔除出去，不再进行比较，这样可以缩短寄存器 S 的长度。简化方法如下：

1）以4.1节中1）为基础，设向量 $\boldsymbol{Y}' = (y_1', y_2', \cdots, y_n')$，若有 $c_{i,j}' = 1$，则令 $y_j' = 1$，它表示 C^i 与硬判决码字 $\boldsymbol{Y}$ 不同的比特；否则，$y_j' = 0$，它表示所有码字此处比特值与硬判决相同，即不存在竞争码字。因此，$r_j = \beta d_j = \beta(2y_j - 1)$。

2）令 $s^+ = (s_1^+, \cdots, s_l^+, \cdots, s_v^+)$，$s^- = (s_1^-, \cdots, s_l^-, \cdots, s_k^-)$，其中 v 为 $\boldsymbol{Y}'$ 中“1”的个数，由于所选用码字为纠错能力为1的码字，因此 $v \leqslant 2^p + p$。同样，将它们中的元素初始化为 $-\infty$。

3）若 $c_{i,l} = 1$ 且 $p(c'_{i,l}) > s_l^+$，则 $s_1^+ = p(c_1')$；若 $c_{i,l} = 0$ 且 $p(c'_{i,l}) > s_l^-$，则 $s_l^- = p(c'_{i,l})$。

4）比较完毕后，可得 $r' = \frac{1}{2}(s^+ - s^-)$，其中值为 $+\infty$ 或 $-\infty$ 的码元表示该位置不存在竞争码字，其外信息的计算由 β 得出；同时，可以得出最佳判决码字的值 $D' = (\mathrm{sign}(r_1'), \cdots, \mathrm{sign}(r_l'), \cdots, \mathrm{sign}(r_v'))$。

5 复杂度和性能分析

对于以 (n,k,δ) 为子码的乘积码，设不可靠位个数 p。若采用 Pyndiah 算法，计算码字欧氏距离时需计算 $2^p \times n$ 次乘法和 $2^p \times n(n-1)$ 次加法；采用相关算法，仅需计算 $2^p \times (n-1)$ 次加法即可；采用改进的相关算法，仅需计算 $2^p \times (k-1)$ 次加法即可（k 为译码后码字与硬判决码字不同的比特位数，最大值为 $\min(2^p + p, n)$）。

此外，由于省去了对竞争码字的搜索，码字的存储空间也大大减小。设量化比特为 q，Pyndiah 算法中，需占用 $2^p \times (n+q)$ bit 的存储空间；相关算法中，仅需占用 $2qn$ bit 的存储空间；改进的相关算法中最多占用 $2q\ \min(2^p + p, n)$ bit 的存储空间。

由以上对算法的叙述可见，在采用改进的相关算法计算软输出时，仅对计算步骤进行化简，没有进行任何的近似。因此，改进算法对性能没有任何影响。以子码为(32,26)和(64,57)的 eBCH 码为例进行 Matlab 仿真，仿真条件为 QPSK 调制，高斯白噪声，不可靠码元个数 $p = 4$，参与仿真的码字个数为10000，仿真图如图2、3。

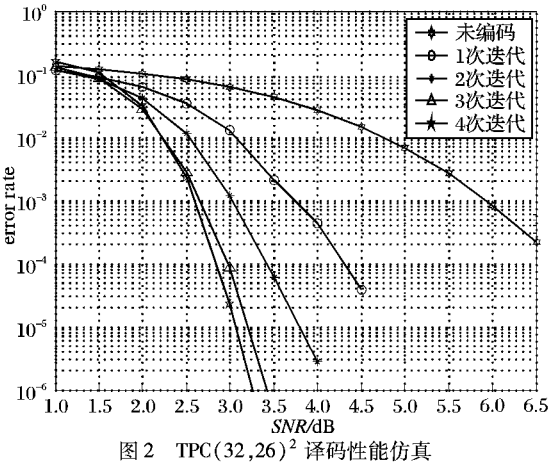

图2 TPC(32,26)^2 译码性能仿真

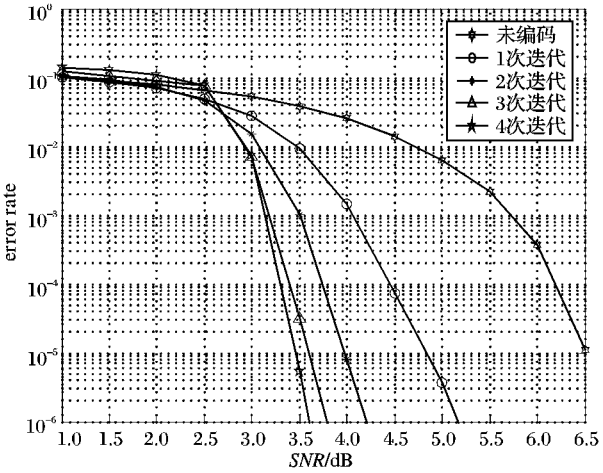

图3 TPC(64,57)^2 译码性能仿真

6 结语

本文介绍了一种基于相关的 TPC 译码算法，并对算法提出了改进，简化了 Chase-Pyndiah 算法，大大降低了译码复杂度，减小了译码延迟。同时，在改进过程中仅采取了较为简单的计算结构，并没有进行任何算法近似，因此得到了与 Chase-Pyndiah 算法相同的译码性能。这种低复杂度算法非常适用于大规模电路的硬件实现。

参考文献：

[1] PYNDIAH R, GLAVIEUX A, PICART A, *et al*. Near optimum decoding of product codes[C]// 1994 IEEE GLOBECOM. New York: IEEE, 1994: 339 - 343.

[2] CHASE D. A class of algorithms for decoding block codes with channel measurement information[J]. IEEE Transactions on Communications, 1972, 18(1): 170 - 182.

[3] PYNDIAH R, COMBELLES P, ADDE P. A very low complexity block turbo decoder for product codes[C]// IEEE Global Telecommunications Conference. New York: IEEE, 1996: 101 - 105.

[4] ARGON C, MCLAUGHLIN S W. A parallel decoder for low latency decoding of turbo product codes[J]. IEEE Communications Letters, 2002, 6(2): 70 - 72.

[5] ELIAS P. Error-free coding[J]. IEEE Transactions on Information Theory, 1954, 4(4): 29 - 37.

[6] VANSTRACEELE C, GELLER B, BROSSIER J M, *et al*. A low complexity block turbo decoder architecture[J]. IEEE Transactions on Communications, 2008, 56(12): 1985 - 1987.

[7] PYNDIAH R. Near-optimum decoding of product codes: Block turbo codes[J]. IEEE Transactions on Communication, 1998, 46(8): 1003 - 1010.