

一种改进的大数逻辑译码算法

张鹏 吴嗣亮

(北京理工大学电子工程系 北京 100081)

摘要: 大数逻辑译码算法的突出优点是实现非常简单,但其纠错能力不强。该文提出了一种改进方法,除利用接收矢量中正确码元提供的信息外,还利用了错误码元提供的信息。改进算法遵循码字错误概率最小和码元错误概率最小两种最佳译码准则,实现了最小距离译码。理论分析和仿真结果均表明改进算法可有效提高纠错能力。

关键词: 大数逻辑译码算法; Reed 算法; Reed-Muller 码

中图分类号: TN911.22

文献标识码: A

文章编号: 1009-5896(2008)05-1264-03

An Improved Majority-Logic Decoding Algorithm

Zhang Peng Wu Si-liang

(Department of Electronic Engineering, Beijing Institute of Technology, Beijing 100081, China)

Abstract: The advantage of majority-logic decoding algorithm is its very simple implementation; however, it provides modest error-correcting capability. In this paper, an improved scheme is proposed, which employs information provided by error symbols as well as correct ones in a received vector. The improved algorithm carries out minimum distance decoding, and complies with two optimal decoding rules, which make the error possibility of code word and the error possibility of code symbol minimum, respectively. Theoretical analysis and computer simulations show that the improvement can enhance efficiently error-correcting capability.

Key words: Majority-logic decoding algorithm; Reed algorithm; Reed-Muller code

1 引言

大数逻辑译码算法^[1]的种类很多,它们均有自己的特点。例如,Reed 算法是从生成矩阵找出正交和式, Massey 算法则是从校验矩阵找出正交和式,而 Rudolph 算法使用的是非正交和式。这些算法的共同点是均采用了择多逻辑判决法则。大数逻辑译码算法实现简单,译码速度快,无需大容量存储器,具有很高的工程实用价值。其缺点主要有两个:一是只能用来译几小类码;二是提供了中等纠错能力,编码增益较低。

大数逻辑可译码虽然种类有限,但并未限制其在现代数字移动通信系统中的广泛应用。例如,WCDMA 系统采用的(32,10)二阶 Reed-Muller(RM)子码^[2]和 TETRA 数字集群系统采用的(30,14)缩短 RM 码都是大数逻辑可译码。由于译码速度快,相信大数逻辑可译码在未来的光纤通信系统中有广泛的应用前景^[3]。

大数逻辑译码算法遵循码元错误概率最小的最佳译码准则,只利用了接收矢量中正确码元提供的信息,而未利用错误码元提供的信息。本文提出了一种能有效提高纠错能力的改进方法,它同时遵循码字错误概率最小和码元错误概率最小两种最佳译码准则,利用了接收矢量中所有码元提供的信息,实现了最小距离译码。本文给出了改进算法详细的译码步骤,并通过仿真验证了其纠错能力的提高。

2 大数逻辑译码算法及其改进

本节首先简单回顾 Reed 算法,然后对其进行改进。改进思想可推广到其它大数逻辑译码算法。

设 (n, k) 二进制线性分组码 C 的最小汉明距离是 d_{\min} , 生成矩阵为 $\mathbf{G} = [\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{k-1}]^T$, 其中 \mathbf{g}_i ($i = 0, 1, \dots, k-1$) 为矩阵行矢量,信息矢量为 $[m_0, m_1, \dots, m_{k-1}]$, 正交于 m_i 的校验和式至多有 J_i 个。Reed 算法的择多逻辑判决法则可叙述如下:若不多于 $\lfloor J_i/2 \rfloor$ ($\lfloor J_i/2 \rfloor$ 表示不大于 $J_i/2$ 的最大整数)个正交校验和式的值为 1, 则 m_i 译成 0; 否则, m_i 译成 1, 并将 \mathbf{g}_i 模 2 加到接收矢量上,以去除 m_i 对接收矢量的贡献。重复上述过程,直至译完全部 k 位信息。令 $J = \min\{J_0, J_1, \dots, J_{k-1}\}$, 则 Reed 算法的纠错能力为 $\lfloor (J-1)/2 \rfloor$ 。

众所周知,码 C 的纠错能力极限为 $\lfloor (d_{\min}-1)/2 \rfloor$, 故 $J \leq d_{\min}$ 。只有当 J 等于或非常接近 d_{\min} 时,Reed 算法对该码才有效。 $J < d_{\min}$ 意味着 Reed 算法不能充分发挥潜在纠错能力,两者的差值越大纠错性能越差。下面对 Reed 算法进行改进,以提高纠错能力。

不失一般性,设 $J_0 \geq J_1 \geq \dots \geq J_{k'-1} \geq \dots \geq J_{k-1}$, 则 $J = J_{k-1}$ 。将信息矢量 $[m_0, m_1, \dots, m_{k-1}]$ 分成 $[m_0, m_1, \dots, m_{k'-1}]$ 和 $[m_{k'}, m_{k'+1}, \dots, m_{k-1}]$ 两部分。与此划分方式相对应,生成矩阵 \mathbf{G} 也分为 $[\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{k'-1}]^T$ 和 $[\mathbf{g}_{k'}, \mathbf{g}_{k'+1}, \dots, \mathbf{g}_{k-1}]^T$ 两部分,前者生成包含 $2^{k'}$ 个矢量的基码空间,后者生成包含 $2^{k-k'}$ 个矢量的掩码空间。基码矢量和掩码矢量的线性组合即为发送矢量。对于一个给定的接收矢量,若掩码矢量是已知

的，亦即掩码对应的信息矢量是已知的，则只要采用 Reed 算法译出基码对应的信息矢量，将译码结果与掩码信息矢量相结合即可恢复完整的信息矢量。既然掩码矢量有 $2^{k-k'}$ 种，发送矢量中的掩码分量必为其中之一，那么恢复的信息矢量有 $2^{k-k'}$ 种可能，在 Reed 算法的纠错能力范围内，信息矢量能被正确恢复，且必为其中之一。于是，问题转化为如何从 $2^{k-k'}$ 种可能中选取一种？这里采用最小距离译码。对所有恢复的信息矢量进行编码，编码结果与接收矢量模 2 加得到 $2^{k-k'}$ 种错误矢量，通过比较选择重量最小的错误矢量对应的信息矢量作为最终的译码结果。以上译码过程即为改进的 Reed 算法，其并行译码结构如图 1 所示。

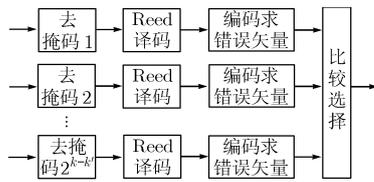


图 1 改进的 Reed 算法的并行译码结构

图 1 采用了并行处理方式，译码速度与原算法相当， $2^{k-k'}$ 个支路的结构完全相同，所需的硬件资源随 k' 的减小成指数增长。改进的 Reed 算法亦可采用串行处理方式，只保留一条支路。一次去除一种掩码矢量，将恢复的信息矢量及其对应的错误矢量存储起来。重复上述操作，直至处理完 $2^{k-k'}$ 种掩码矢量，然后进行比较选择输出。这种方式虽可减少硬件资源，但会不可避免地降低译码速度。因此，实现时需要在译码速度和纠错能力之间加以折衷。

观察图 1 不难发现，改进算法的纠错能力由 Reed 译码和比较选择两个功能单元共同决定。令 $J' = \min\{J_0, J_1, \dots, J_{k'-1}\} = J_{k'-1}$ ，则 Reed 译码功能单元的纠错能力为 $\lceil (J' - 1)/2 \rceil$ ，而比较选择功能单元恰恰等于码的纠错能力极限。因此，改进算法的纠错能力为 $\lceil (\min\{J', d_{\min}\} - 1)/2 \rceil$ 。当 $J = d_{\min}$ 时，改进算法徒增计算复杂度而不会提高纠错能力。当 $J < d_{\min}$ 时，可分 3 种情况讨论：

- (1) $J' = J$ 。纠错能力不会得到提高。
- (2) $J < J' < d_{\min}$ 。在这种情况下， J' 越大纠错能力增加越大。

(3) $J' \geq d_{\min}$ 。纠错能力达到极限 $\lceil (d_{\min} - 1)/2 \rceil$ 。此时， k' 的取值越大越好，这是因为 k' 的取值越小意味着要么需要更多的硬件资源，要么造成译码速度更慢。

众所周知，码字错误概率最小和码元错误概率最小是应当遵循的两个最佳译码准则。Reed 算法是利用接收矢量中正确码元提供的信息恢复信息矢量，遵循的是后者。改进算法则是一方面利用正确码元提供的信息恢复部分信息位，遵循的是后者；另一方面利用错误码元提供的信息计算错误矢量，通过比较选择重量最小的错误矢量恢复其余信息位，遵

循的是前者。总而言之，改进算法充分地利用了接收矢量提供的信息，综合考虑了两种最佳译码准则，实现了最小距离译码，使纠错能力得到提高。

3 仿真结果与分析

为了比较大数逻辑译码算法及其改进的纠错能力，对 WCDMA 系统的物理层协议中采用的(32,10)二阶 RM 子码^[4]进行了仿真。该码的生成矩阵为

$$G = \begin{bmatrix} 1010101010101010101010101010100 \\ 01100110011001101100110011001100 \\ 00011110000111100011110000111100 \\ 0000000111111110000001111111100 \\ 0000000000000001111111111111101 \\ 11111111111111111111111111111111 \\ 0101000011000111110000011011101 \\ 0000001110011011101101110001100 \\ 00010101111100100110110010101100 \\ 0011100001101110101110101000100 \end{bmatrix}$$

通过计算机搜索^[5]很容易得到正交于各个信息位的和式组，具体如下：

- $m_0 : 0,1,11,14;5,6,16,17;10,13,15,18;19,20,23,24;4,8,21,25;7,9,26,28;2,3,22,29;12,27,30,31$
- $m_8 : 1,3,8,14;4,6,15,17;0,2,19,21;5,7,20,22;16,18,24,26;9,13,23,27;11,12,28,29;10,25,30,31$
- $m_7 : 2,6,7,11;0,1,12,13;4,9,19,24;16,21,23,26;8,14,25,27;5,10,17,28;15,20,22,29;3,18,30,31$
- $m_6 : 0,1,15,16;7,9,18,20;5,12,17,22;8,10,23,25;2,4,24,26;3,13,21,27;6,11,19,28;14,29,30,31$
- $m_4 : 0,15;1,16;2,17;3,18;4,19;5,20;6,21;7,22;8,23;9,24;10,25;11,26;12,27;13,28;14,29;30,31$
- $m_3 : 0,8;1,9;2,10;3,11;4,12;5,13;6,14;15,23;16,24;17,25;18,26;19,27;20,28;21,29;7,30;22,31$
- $m_2 : 0,4;1,5;2,6;7,11;8,12;9,13;10,14;15,19;16,20;17,21;22,26;23,27;24,28;25,29;3,30;18,31$
- $m_1 : 0,2;3,5;4,6;7,9;8,10;11,13;12,14;15,17;18,20;19,21;22,24;23,25;26,28;27,29;1,30;16,31$
- $m_0 : 1,2;3,4;5,6;7,8;9,10;11,12;13,14;16,17;18,19;20,21;22,23;24,25;26,27;28,29;0,30;15,31$
- $m_5 : 0;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20;21;22;23;24;25;26;27;28;29;30;31$

译码时要按照从上到下的顺序。需要说明的是，在上面的表示形式中，两个正交和式之间用分号隔开，用逗号隔开的十进制数字表示接收矢量中元素的下标。既然 $J_9 = J_8 = J_7 = J_6 = 8$ ， $J_4 = J_3 = J_2 = J_1 = J_0 = 16$ ， $J_5 = 32$ ，那么 $J = 8$ 。已知 $d_{\min} = 12 > J$ ，显然采用 Reed 算法会牺牲该码的潜在纠错能力。

将生成矩阵 \mathbf{G} 分为 $[g_0, g_1, \dots, g_5]^T$ 和 $[g_6, g_7, \dots, g_9]^T$ 两部分, 它们分别生成基码空间和掩码空间。在这种划分方式下, Reed 译码功能单元译低 6 位信息, 高 4 位信息由比较选择功能单元译出。因为 $J' = 16 > d_{\min}$, 所以改进算法提高了纠错能力, 达到了极限。

在 WCDMA 系统中, 32 位码元在某些情况下并不全都传输, 而是舍弃若干高位只传输剩余低位。码长的缩短会导致最小汉明距离变小, 正交和式的最少数目变小, 如表 1 所示。在各种码长下, $J < d_{\min}$, Reed 算法的性能均不理想; 而 $J' > d_{\min}$, 改进算法始终能达到纠错能力的极限。

表 1 不同码长下的 d_{\min} , J 和 J'

码长	25	26	27	28	29	30	31	32
d_{\min}	7	7	7	8	9	10	11	12
J	3	3	4	5	6	7	7	8
J'	9	10	11	12	13	14	15	16

选取码长 $n = 30$ 和 $n = 32$ 两种典型情况, 对 Reed 算法及其改进进行仿真。图 2 给出了未编码系统和 4 种编码系统在加性高斯白噪声信道(AWGN)下的误码性能, 横坐标表示信噪比(SNR), 纵坐标表示误码率(BER)。从图中可以看出, 在两种码长下, 改进算法的纠错性能均明显优于原算法; 采用相同的译码算法, 长码的性能稍微优于短码, 这是因为前

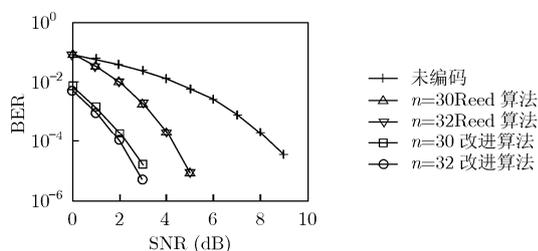


图 2 未编码系统和 4 种编码系统的误码性能

者具有更多的冗余信息。当误码率为 10^{-4} 时, $n = 30$ 和 $n = 32$ 的改进算法分别提供了大约 6.1dB 和 6.3dB 的编码增益, 分别比原算法提高了 2dB 和 2.2dB 左右。

4 结束语

本文以 Reed 算法为例提出了一种能有效提高纠错能力的改进大数逻辑译码算法。改进算法遵循码字错误概率最小和码元错误概率最小两种最佳译码准则, 合理地利用了接收矢量中所有码元提供的信息, 实现了最小距离译码, 而不像遵循码元错误概率最小一种最佳译码准则的原算法那样, 只利用正确码元提供的信息。仿真结果表明, 改进算法较原算法能极大降低误码率, 提高纠错性能。

参考文献

- [1] Lin S and Costello D J. Error control coding: fundamentals and applications. New Jersey: Prentice-Hall Inc., 1993: 184-257.
- [2] 张丙杰, 胡捍英, 刘正军. WCDMA通信系统中传输格式字分配策略及译码. 微计算机信息, 2005, 21(4): 186-187.
Zhang B J, Hu H Y, and Liu Z J. Assigning and decoding of TFCI in WCDMA system. *Control & Automation*, 2005, 21(4): 186-187.
- [3] Wicker S B. Error control systems for digital communications and storage. New Jersey: Prentice Hall, Inc., 1995: 149-150.
- [4] ETSI. 3GPP TS 25.212 V7.2.0. ETSI, 2006.
- [5] 陈瑾, 王金龙. 一种改进的Reed译码算法. 通信学报, 2000, 21(11): 82-85.
Chen J and Wang J L. An improved Reed decoding algorithm. *Journal of China Institute of Communications*, 2000, 21(11): 82-85.

张 鹏: 男, 1978 年生, 博士生, 研究方向为集群移动通信、数字信号处理等。

吴嗣亮: 男, 1964年生, 教授, 博士生导师, 主要研究方向为现代信号处理的理论与应用、雷达系统理论与技术、电子系统仿真与信号模拟等。