

自适应量化测试序列数的分组 Turbo 码译码算法

刘星成 王康

(中山大学电子与通信工程系 广州 510275)

摘要: 针对分组 Turbo 码自适应 Chase 译码算法存在的缺陷, 该文提出自适应量化测试序列数的分组 Turbo 码译码算法。该方法以测试序列数 C 为研究对象, 依出错概率大小选择错误图样, 并利用量化测试函数根据 SNR 的变化对测试序列数进行量化, 从而达到直接控制译码复杂度的目的。仿真结果表明, 所提出的译码算法保证了译码性能, 并直接降低了译码复杂度。

关键词: 分组 Turbo 码; 测试序列数; Chase 算法; 信噪比; 误码率

中图分类号: TN911.22

文献标识码: A

文章编号: 1009-5896(2009)12-3006-04

Decoding Algorithm for Block Turbo Codes Based on the Adaptive Quantized Testing Sequences

Liu Xing-cheng Wang Kang

(Department of Electronic and Communications Engineering, Sun Yat-sen University, Guangzhou 510275, China)

Abstract: Considering the defects in the adaptive Chase algorithm, a novel decoding algorithm for block Turbo codes based on the adaptive quantized testing sequences is proposed. This algorithm is employed to investigate the number of testing sequences, select the test patterns according to the possibility of least reliable bits, and use the quantizing-testing function to quantize the testing sequences according to the level of SNR, which can adaptively adjust the complexity of decoding. The simulation results show that, compared with the traditional algorithms, the proposed algorithm can reduce the decoding complexity with the same BER performance.

Key words: Block Turbo Codes (BTC); Testing sequences; Chase algorithm; Signal-to-Noise Ratio (SNR); Bit Error Rate (BER)

1 引言

1993 年, Berrou 等人首次提出了一种新型信道编码方案——Turbo 码^[1,2], 该码能获得几乎接近 Shannon 理论极限的译码性能。1994 年, Pyndiah 提出了一种基于分组码的编码和译码算法, 称之为分组 Turbo 码(BTC)^[3,4], 并显示分组 Turbo 码能够达到与卷积 Turbo 码相媲美的性能。分组 Turbo 码在性能与复杂度之间提供了较好的平衡, 降低了译码复杂度, 实现也比较简单, 但也仍然存在一定缺陷。

Mahran 和 Weber 等人也提出了降低译码复杂度的方法, 并给出了理论推导和性能分析^[5-8]。Zhou 等人^[9,10]提出了改进的 AD-Chase 算法, 较好地降低了译码复杂度。但是, AD-Chase 算法是在假定各不可靠比特位出错概率相等的前提下, 形成相同数量的不可靠比特数来间接降低译码复杂度, 而实际

中, 这种假设各比特位出错概率相同并固定不可靠比特数的方法存在一定问题, 因此有必要对该 Chase 算法做进一步改进。

借鉴量化测试函数对模拟量量化的原理, 本文提出自适应量化测试序列数的分组 Turbo 码译码算法: 优先选取出错概率高的错误图样, 并通过量化测试函数根据 SNR 变化自适应调整译码复杂度。仿真结果表明, 自适应量化测试序列数的分组 Turbo 码译码算法能够在降低译码复杂度, 缩短译码时间的同时, 获得较好的译码性能。

2 分组 Turbo 码 AD-Chase 译码算法

Mahran 和 Benaissa, 对 Chase 算法做了改进, 并提出了分组 Turbo 码 AD-Chase 译码算法^[5]。其改进如下:

(1) 根据信道噪声的情况以及码率, 设定门限,

$$T(\gamma) = A(\gamma + 1) \sqrt{\left(\frac{k}{n}\right) \left(\frac{E_b}{N_0}\right)} \quad (1)$$

其中 A 是量化因子, 一般取常数, γ 是置信因子, 其值随着迭代而变化, k/n 是码率。

2008-12-31 收到, 2009-06-29 改回

国家自然科学基金(60673086, 60711140419), 广东省科技计划(2006B50101003)和广州市科技计划(2007Z3-D0071)资助课题

(2)依据可靠度,得到不可靠比特。对于接收矩阵某一行/列中处于不可靠门限函数内的比特,依可靠度排序,应用式(2)将门限函数内的 P 个不可靠比特判决为 e^* ,并对门限函数外的比特利用式(3)提前判决为 0 或 1,其中 δ 为码字的最小欧氏距离。

$$x_j = \begin{cases} 0, & \alpha_j < -T \\ e^*, & -T \leq \alpha_j < T, \quad j = 1, 2, \dots, P \\ 1, & \alpha_j \geq +T \end{cases} \quad (2)$$

$$x_j = \begin{cases} 0, & \alpha_j < 0 \\ 1, & \alpha_j > 0 \end{cases}, j = P + 1, P + 2, \dots, n \quad (3)$$

(3)通过错误图样形成竞争码字集空间及软判决码字序列 D 。错误图样分别加上由接收矩阵 R 中对应行/列经过解调得到的硬判决序列 Y ,形成测试序列集后,分别送入译码器得到 2^P 个竞争码字集,再将竞争码字集中与接收矩阵 R 中对应行/列具有最小欧氏距离的码字作为软判决码字 D 。

(4)计算软判决码字外信息。逐位搜索竞争码字集中满足 $d_j \neq c_j$,并具有次最小欧式距离的码字作为竞争码字 C ,并计算软判决码字 D 中 d_j 位的外信息。

如果能够找到 C ,则对于 d_j 位的可靠度计算公式为^[4]

$$r'_j = \left(\frac{|R - C|^2 - |R - D|^2}{4} \right) d_j \quad (4)$$

如果不能找到竞争码字 C ,则可靠度计算公式为^[4]

$$r'_j = \beta d_j \quad (5)$$

其中 β 为可靠度因子 ($\beta > 0$),由此可得到对应于 d_j 位的外信息为 $w_j = r'_j - r_j$ 。

(5)外信息与接收矩阵 R 进行多次迭代得到最优码字矩阵。将外信息序列与权重因子 $\alpha(2(m-1)+1)$ 或 $\alpha(2(m-1))$ 相乘,并与接收矩阵 R 的对位相加后,得到 $m+1$ 次全迭代译码中对应位的接收序列 R 。

下面分析 AD-Chase 译码算法所存在缺陷。实验表明^[5],在 BER=10⁻⁵ 时,AD-Chase 译码算法能够降低 15%~60% 的运算复杂度。但是,这种译码算法还存在如下缺陷:

(1)AD-Chase 算法中 SNR 为估计值,这意味门限函数将存在误差,这将极大影响译码复杂度。表 1 数据表明,SNR 估计值与实际 SNR 有一定的误差,这将影响门限函数的估计,并最终影响译码效果和译码复杂度。

(2)AD-Chase 译码算法,是在假设各个不可靠比特出错概率相等的前提下,每次迭代生成固定数目的不可靠比特数。而事实上,由于各个不可靠比特的可靠度不同,由不可靠比特组成的错误图样的出现概率也存在差异。

例如,3 个不可靠比特,出错概率分别为(0.4,0.3,0.2),则有错误图样(000)的出现概率为 000→(1-0.4)×(1-0.3)×(1-0.2)=0.336;错误图样(111)的出现概率为 111→(0.4)×(0.3)×(0.2)=0.024。

又由表 2 可知,每行/列不可靠比特数 P 对译码复杂度影响较大,因此这种每次迭代产生固定数目的不可靠比特数的做法,一定程度上浪费了译码时间。

(3)AD-Chase 译码算法中,没有根据 SNR 的变化调整所需要的测试序列数,这造成了运算的浪费。表 3 为在译码性能达到同一数量级时,固定测试

表 1 SNR 估计值与 SNR 实际值的比较(总数据量为 9.61×10⁶)

SNR 实际值(dB)	1.0	1.5	2.0	2.5	3.0
SNR 估计值(dB)	0.907645	1.20018	2.04115	2.57727	3.08900

表 2 每行/列平均不可靠比特数 P 对分组 Turbo 码译码复杂度的影响

译码步骤	依可靠度排序	生成测试序列	硬判决译码	计算欧氏距离	欧氏距离排序	寻找竞争码字	计算外信息
加减法次数	$n^2(n-1)$	$P \times 2n \times 2^P$	$2n^2(n-k) \times 2^P$	$2^P \times 2n^2$	$n \times 2^P \times (2^P - 1)$	$2n^2 \times 2^P$	n^2

表 3 (32,26,4)³BTC 译码时平均不可靠比特 P 比较(迭代 5 次,数据量为 9.61×10⁶)

SNR	1.0	1.5	2.0	2.5	3.0
BER	10 ⁻³	10 ⁻³	10 ⁻⁴	10 ⁻⁴	10 ⁻⁵
固定 P	3	3	3	3	3
浮动 P	2.65071	2.35299	2.08010	1.93164	1.682265

序列与浮动测试序列的平均复杂度比较。因此,应当在信噪比较低时,译码器需要较多的测试序列来形成竞争码字,随着信噪比的增加,译码器不再需要过多的测试序列就能够得到较好的误码率。

3 自适应量化测试序列数的译码算法

借鉴通信中对模拟量进行量化的方法,改变 AD-Chase 译码算法中步骤(1),步骤(2),步骤(3),提出自适应量化测试序列数的译码算法:以形成错误图样的测试序列数 C 作为研究对象,将错误图样依照出错概率大小排序,根据 SNR 的变化通过量化测试函数确定测试序列的数量,优先选择出错概率高的错误图样。对 BTC(n,k,t)² 码译码,改进的 Chase 算法译码按照下列步骤进行:

(1)先找出 $P=[\delta/2]$ 个最不可靠比特,其中 δ 为最小汉明距离。根据解调器输出的接收序列 R 得到硬判决序列 Y ,并对接收序列 R 的可靠性按从小到大进行排序,从中找出 $P=[\delta/2]$ 个不可靠比特。

(2)生成 2^P 个错误图样,并依据出错概率从大到小排列。例如: $P=3$ 时,错误图样为(000, 100, 010, 110, 001, 101, 011, 111)。

(3)按照出错概率大小顺序依据量化测试函数 $C = [f[(E_b/N_0) \times R]]$ (其中 $[f(\cdot)]$ 为取 $f(\cdot)$ 的整数部分),从错误图样集合中找出 C 个错误图样,并分别加上硬判决序列 Y ,形成测试序列集合后,最终选出与接收矩阵 R 中对应行/列具有最小欧氏距离的码字作为软判决码字。

(4)与 AD-Chase 算法相同,通过逐位搜索竞争码字集,得到竞争码字 C ,并计算软判决码字 D 中 d_i 位的外信息,并将外信息与接收矩阵 R 进行多次迭代,最终得到最优码字矩阵。

4 量化测试函数的选取及仿真实验

表 3 实验结果表明,同数量级的译码性能,AD-Chase 算法所实际需要的不可靠比特数 P 随着 SNR 的增加而减小。低 SNR 值时,译码器需要较多的测试序列数来形成竞争码字,反之,高 SNR 值时则较少的竞争码字就能保证译码性能。因此,在同数量级译码性能时,测试序列函数总体趋势应与 AD-Chase 算法所需不可靠比特数 P 趋势相同:随 SNR 递增而逐级递减。

设定量化测试函数 $C = [f[(E_b/N_0) \times R]]$,其中, $C=[f(x)]$ 为取整函数; $1 \leq f(x) \leq (2^p=8)$; $0 \text{ dB} \leq x \leq 4 \text{ dB}$; $R=k/n$ 为码率; E_b/N_0 为归一化信噪比; $f(x)$ 为在不同 SNR 值,在与 AD-Chase 算法同数量级译码性能时,所需的平均测试序列数 C 。

在试验中, $f(x)$ 是通过实验数据描点连线获得的 (SNR, C) 曲线,并以折线函数近似逼近得到的。以 BTC(32,26,4)² 码为例,为得到同级别译码性能,用 4 种不同量化测试函数近似逼近表 3 实验结果形成的 (SNR, C) 曲线。

$$f_1(x) = \begin{cases} 8, & x \leq 0 \\ -2x + 8, & 0 < x < 4 \\ 1, & x \geq 4 \end{cases} \quad (6)$$

$$f_2(x) = \begin{cases} 8, & x \leq 0 \\ -(4/3)x + 8, & 0 < x < 4 \\ 1, & x \geq 4 \end{cases} \quad (7)$$

$$f_3(x) = \begin{cases} 8, & x \leq 0 \\ -2x + 6, & 0 < x < 2 \\ -x + 6, & 2 \leq x \leq 4 \\ 1, & x > 4 \end{cases} \quad (8)$$

$$f_4(x) = \begin{cases} 8, & x \leq 0 \\ -4/3x + 6, & 0 < x \leq 3 \\ -3/2x + 6, & 3 < x \leq 4 \\ 1, & x > 4 \end{cases} \quad (9)$$

由上述 4 个量化测试函数得到的译码性能如图 1 所示,实验所用调制方式为 BPSK,传输信道为 AWGN。当 $1 \text{ dB} \leq E_b/N_0 \leq 2.75 \text{ dB}$ 时,译码性能依 $f_4(x)$ 、 $f_2(x)$ 、 $f_3(x)$ 、 $f_1(x)$ 的顺序降低。而当 $E_b/N_0=4 \text{ dB}$ 时,性能依 $f_3(x)$ 、 $f_2(x)$ 、 $f_4(x)$ 、 $f_1(x)$ 的顺序降低。

图 2 为 4 种量化测试函数对于 BTC(32, 26, 4)² 码行/列平均硬判决次数的影响,表 4 为 BTC (32, 26, 4)² 译码中不同量化测试函数算法对程序运行时间的影响,以及与 AD-Chase 算法的比较。比较图 2 及表 4 可得,总体上译码复杂度依 $f_1(x)$ 、 $f_3(x)$ 、 $f_4(x)$ 、 $f_2(x)$ 的顺序提升。

因此,综合译码性能以及译码复杂度,选取 $f_4(x)$ 作为 4 种曲线的最优量化测试函数,得到的译码性能曲线如图 3 所示。在迭代 4 次, $E_b/N_0=3 \text{ dB}$ 时, Maharan 等人提出的 AD-Chase 算法, (32, 26, 4)² 码以 0.5 dB 的性能损失为代价,换取了降低 60% 译码复杂度的结果^[5]。综合表 4、图 2 和图 3 可知,本文所提出的自适应量化序列数译码算法,以 0.25 dB 为代价降低 70% 译码复杂度,且译码速度明显提升。由此可见本文所提出的算法的优势。

5 结论

本文提出的自适应量化测试序列数的译码算

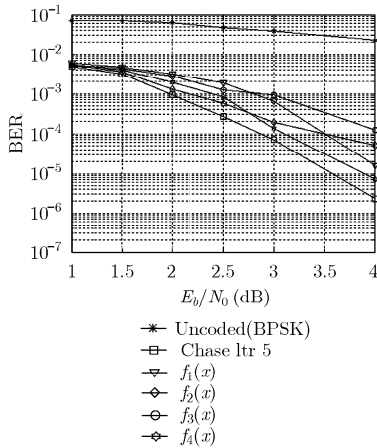


图1 BTC(32,26,4)²码取不同量化测试函数译码时的性能

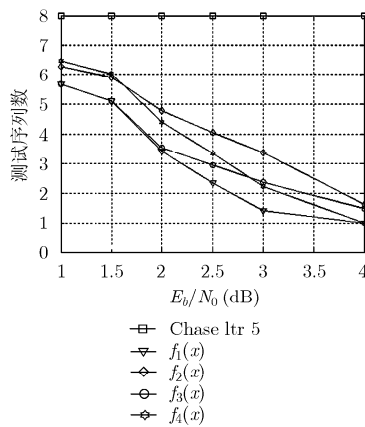


图2 不同量化测试函数对于 BTC(32,26,4)²码行/列平均硬判决次数的影响

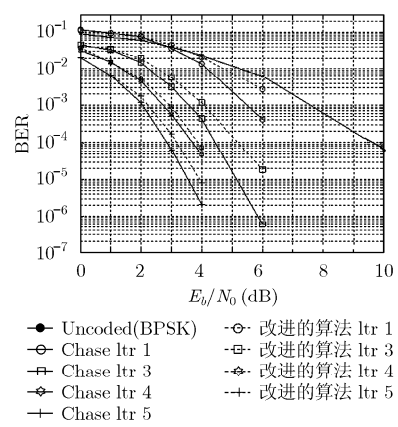


图3 BTC(32,26,4)²码自适应量化测试序列数译码算法($f_i(x)$)性能曲线

表4 (32,26,4)² BTC 译码中不同算法对程序运行时间的影响(迭代5次, 数据量 9.61×10^6)

SNR(dB)	程序总运行时间(s)				
	Chase	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
1.0	398.47	342.43	362.92	341.28	371.11
1.5	393.73	303.29	322.43	303.73	335.54
2.0	396.57	251.91	291.62	254.57	285.12
2.5	394.75	207.76	279.17	225.12	256.71
3.0	397.75	155.84	236.82	187.29	175.29
4.0	396.82	151.97	161.04	157.28	152.68

法, 直接以测试序列数作为参考对象, 通过改变量化测试函数来直接改变译码复杂度, 同时改进译码性能, 本算法具有以下特点:

(1)可以直接控制译码复杂度与译码性能。由于该算法直接改变参与形成竞争码字的测试序列数, 因此改变量化测试函数可以自由调整译码复杂度与译码性能。相对于 AD-Chase 算法, 该算法实用性强, 应用灵活。

(2)降低译码复杂度。在迭代4次, $E_b/N_0=3$ dB 时, BTC(32,26,4)² 码的自适应量化序列数译码算法以 0.25 dB 为代价, 降低了 70%译码复杂度。与 AD-Chase 算法相比, 所提出的自适应量化测试序列数译码算法能够以更小的性能损失换取更多运算复杂度的降低, 因此性能更具优势。

参考文献

[1] Berrou C, Glavieux A, and Thitimajshima P. Near Shannon limit error-correcting coding and decoding: Turbo-codes. ICC'93, Geneva, May 1993, 2: 1063-1070.
 [2] Berrou C and Clavieux A. Near optimum error correcting coding and decoding: turbo-codes. *IEEE Transactions on Communications*, 1996, 44(10): 1261-1271.
 [3] Pyndiah R, Glavieux A, and Picart A, et al. Near optimum decoding of product codes, GLOBECOM' 94, San Francisco, CA, 28 Nov-2 Dec. 1994: 339-343.

[4] Pyndiah R. Near-optimum decoding of product codes: Block Turbo codes. *IEEE Transactions on Communications*, 1998, 46(8): 1003-1010.
 [5] Mahran A and Benaissa M. Adaptive Chase algorithm for block Turbo codes. *Electronics Letters*, 2003, 39(7): 617-619.
 [6] Weber J H and Fossorier M P C. Limited-trial Chase-like algorithms achieving bounded-distance decoding. *IEEE Transactions on Information Theory*, 2004, 50(12): 3318-3323.
 [7] Xu C, Liang Y C, and Leon W S. Shortened Turbo product codes: encoding design and decoding algorithm. *IEEE Transactions on Vehicular Technology, Part 1*, 2007, 56(6): 3495-3501.
 [8] Liu W, Xiong C, and Wang D, et al. A QRD-based chase decoding algorithm for quasi-orthogonal space-time block code systems. WiCOM'08, Dalian, China, Oct. 2008: 1-5 .
 [9] Zhou R, Le Bidan R, and Pyndiah R, et al. Low-complexity high-rate Reed-Solomon block Turbo codes. *IEEE Transactions on Communications*, 2007, 55(9): 1656-1660.
 [10] Loncar M, Johannesson R, and Bocharova I E, et al. Soft-output BEAST decoding with application to product codes. *IEEE Transactions on Information Theory*, 2008, 54(3): 1036-1049.

刘星成: 男, 1964年生, 副教授, 研究方向为纠错编码和无线通信。
 王康: 男, 1982年生, 工程师, 研究方向为纠错编码。