

基于自适应估计 SNR 的分组 Turbo 码译码算法*

刘星成, 王 康, 黄志军
(中山大学 电子与通信工程系, 广东 广州 510275)

摘 要: 针对分组 Turbo 码自适应 Chase 译码算法中 SNR 是预先设定值的情况, 提出一种新的自适应估计信噪比(SNR)的译码算法。该方案利用了接收码字的统计信息与 SNR 之间存在的对应关系来调整门限函数, 达到控制译码复杂度的目的。仿真结果表明, 所提出的译码算法能自适应反映信道情况, 降低译码运算复杂度, 提高译码处理速度, 获得了较好的性能。

关键词: 分组 Turbo 码; 迭代译码; Chase 算法; 信噪比(SNR); 比特差错率(BER)

中图分类号: TN911.22 **文献标识码:** A **文章编号:** 0529-6579(2009)05-0033-05

Decoding Algorithm for Block Turbo Codes Based on Adaptive SNR Estimation

LIU Xingcheng, WANG Kang, HUANG Zhijun
(Department of Electronic and Communications Engineering,
Sun Yat-sen University, Guangzhou 510275, China)

Abstract: Considering the fact that the SNR values in the adaptive Chase algorithm are prefixed in the literatures, a modified decoding algorithm for block Turbo codes was proposed, in which the level of the channel SNR was adaptively estimated. The proposed algorithm could tune the decoding complexity with the aid of the relationship between the statistics of the received vectors and the channel SNR to change the threshold function. Simulation results show that the proposed algorithm decreases the decoding complexity, speeds up decoding while achieving better BER performance. Furthermore, the proposed decoding algorithm can adapt to different SNR values without prefixing them.

Key words: block Turbo codes (BTC); iterative decoding; Chase algorithm; Signal-to-Noise Ratio (SNR); bit error rate (BER)

Berrou 等人^[1-2]于 1993 年提出了 Turbo 码, 该码能获得接近 Shannon 理论极限的译码性能。1994 年, Pyndiah 等人^[3-5]对分组 Turbo 码(BTC)进行了研究, 但其译码复杂度较高^[6]。有鉴于此, Weber 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^[7-9]。Chi 等人^[10]采用正切函数代替原来的归一化因子, 改善了译码性能。

Armand 和 Mahran 等人针对这个问题提出了自适应 Chase 算法^[6,11-12]。这种算法需要预先设定

SNR 的值, 但预设的值不能反映实际信道情况, 导致门限函数估计不准, 降低译码性能。因此, 对 BTC 码的 Chase 算法需要做进一步改进。与 Summers 等人^[13]的研究类似, 本文借鉴了卷积 Turbo 码中自适应估计 SNR 的方法, 采用外信息方差之差来估计 SNR, 从而达到控制迭代次数的目的。

1 BTC 码的自适应 Chase 算法译码

BTC 码的译码过程如图 1 实线部分所示, 其中

* 收稿日期: 2009-01-09

基金项目: 国家自然科学基金资助项目(60673086, 60970041); 广东省科技计划资助项目(2006B50101003)

作者简介: 刘星成(1964年生)男, 博士, 副教授; E-mail: isslxc@mail.sysu.edu.cn.

调整置信因子 γ 来改变门限函数的。在迭代译码中, 随着外信息的变化, 送入行/列译码器的 R [m] 也在不断变化。因为迭代译码实际上就是通过外信息将原先不可靠的比特修正为可靠比特的过程。送入分组 Turbo 码的接收矩阵 R 的可靠度变化趋势是由低可靠度向高可靠度变化的。因此, SNR 值也会由低信噪比向高信噪比变化, 这样使得接收矩阵中的实际不可靠比特的数目也在不断变化。

对于 BTC 情况, 我们的试验结果表明: 每迭代译码一次, 接收矩阵 R 的可靠度就改进一些, SNR 估计值也由低到高在不断变化。据此, 在估计 SNR 的自适应 Chase 算法的基础上, 提出每次迭代自适应估计 SNR 的 Chase 算法。与自适应估计 SNR 的 Chase 算法不同的是, 该算法不仅第一次迭代时估计 SNR, 而且以后每次迭代也进行图 1 虚线部分的 SNR 估计, 用更新的 SNR 值去改变门限函数, 以进一步降低译码复杂度。

3 仿真及试验结果分析

本实验环境为 Celeron 1.5 GHz 处理器, 512 MB 内存, Windows XP, Visual C++ 6.0。若无特别说明, 所用调制方式为 BPSK, 信道为 AWGN。在仿真实验中, 采用的分组 Turbo 码是由两个 BCH (31, 26, 3) 非系统码组成的。其编码方法分为四步: 将信息序列排列成 26×26 矩阵; 再用 (31, 26, 3) 码对信息矩阵各行编码, 生成 26×31 矩阵; 接着用 (31, 26, 3) 码对信息矩阵各列编码, 生成 31×31 矩阵; 最后将编码序列输出。

对于分组 Turbo 码的行/列译码, 硬判决译码器采用的是梅吉特译码器。在采用 Chase-2 算法的分组 Turbo 码译码中, 每行/列选出 $P=3$ 个不可靠比特, 共产生 $2^3=8$ 个测试序列送入译码器。在自适应估计 SNR 的 Chase 算法中进行分组 Turbo 码译码时, 每行/列最多选出 $P=3$ 个不可靠比特。在采用每次迭代自适应估计 SNR 的 Chase 算法中, 每行/列最多也是选出 $P=3$ 个不可靠比特。

3.1 自适应估计 SNR 的 Chase 算法

自适应估计 SNR 的 Chase 算法, 就是在第一次迭代译码之前, 通过接收矩阵 R 的统计量对 SNR 进行近似估计, 将 SNR 的估计值应用于分组 Turbo 码的自适应 Chase 算法中。图 2 是 (31, 26, 3)² 分组 Turbo 码自适应估计 SNR 的 Chase 算法与 Chase-2 算法的性能比较图。

可以看到, 自适应估计 SNR 的 Chase 算法与 Chase-2 算法在 BER 译码性能上几乎相同。但是,

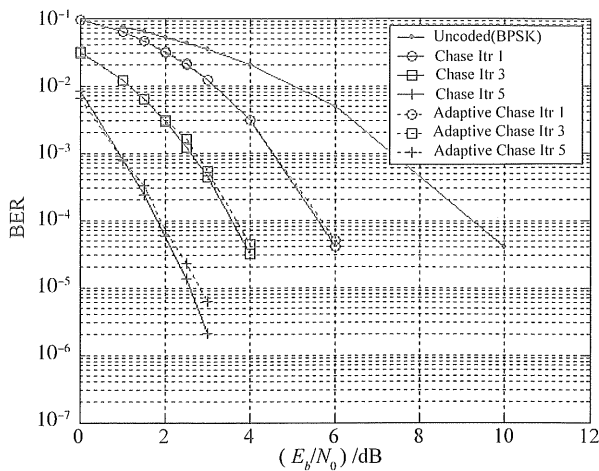


图2 自适应估计 SNR 的 Chase 译码算法 ($A=1$) 与 Chase-2 译码算法性能比较

Fig. 2 Performance comparison between decoding algorithms - adaptive estimating SNR and Chase-2

前者能够自适应估计 SNR, 从而自适应调整译码复杂度。在实验中发现, 改变门限函数中量化因子 A , 可以调整 SNR 估计值在门限函数中的作用, 从而达到改变乘积码译码的平均不可靠比特数的目的。同时, 适当的门限函数能够将门限内的比特判为不可靠比特, 门限外的比特作为可靠比特提前做出判决, 这不仅有助于减小平均不可靠比特数从而降低译码复杂度, 而且能够提升译码性能。

表 1 示出了 (31, 26, 3)² 分组 Turbo 码在量化因子 A 取不同值时的平均不可靠比特数。从该表可知, 当 $A=1$ 时, 门限函数几乎没有起到限定不可靠比特的作用, 接收矩阵 R 中每行/列中的平均不可靠比特数也不随 SNR 的变化而变化。当 $A=2/(SNR \times 10)$ 时, 门限函数随着信噪比的增大而减小, 使得不可靠比特数随信噪比增大而减小。

表 1 量化因子 A 的大小与平均不可靠比特数之间的关系 (迭代 4 次, 数据量为 9.61×10^8)

Table 1 Relationship between the value of quantized factor A and the average No. of unreliable bits (9.61×10^8 Bits at Iteration 4)

SNR/dB	$A=1$	$A=1/$ (SNR $\times 10$)	$A=2/$ (SNR $\times 10$)	$A=3/$ (SNR $\times 10$)
1.0	3	1.485 62	2.267 76	2.681 95
1.5	3	1.012 73	1.776 54	2.319 75
2.0	3	0.774 855	1.420 25	1.962 15
2.5	3	0.618 487	1.153 34	1.644 01
3.0	3	0.497 139	0.939 637	1.371 56

观察表 1 还发现, $A = 2/(\text{SNR} \times 10)$ 时, 其译码复杂度小于 $A = 3/(\text{SNR} \times 10)$, 而整体译码性能又优于 $A = 1/(\text{SNR} \times 10)$ 的情况。因此, $A = 2/(\text{SNR} \times 10)$ 可以较好地平衡运算复杂度与译码性能之间的矛盾。因此, 选取 $A = 2/(\text{SNR} \times 10)$, 得到自适应估计 SNR 的 Chase 算法译码性能, 如图 3 所示。由该图可知, 第 1 次迭代时, Chase - 2 算法的译码性能优于自适应估计 SNR 的 Chase 算法。第 3 次迭代时, 两算法不相上下。第 5 次迭代时, 自适应估计 SNR 的 Chase 算法的译码性能优于 Chase 算法。在迭代 5 次, $E_b/N_0 = 3 \text{ dB}$ 时, 其 BER 可以达到 6×10^{-7} 。

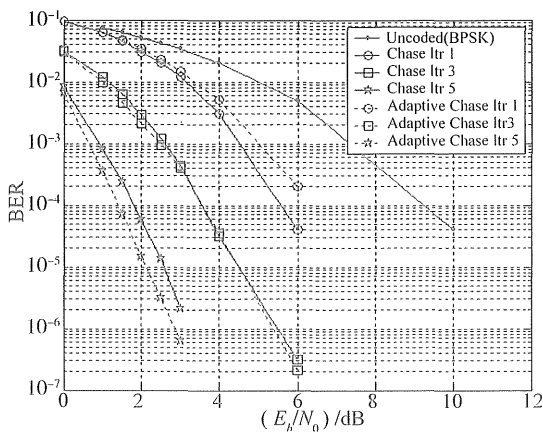


图 3 自适应估计 SNR 的 Chase 算法 (Adaptive Chase) 性能 ($A = 2/(\text{SNR} \times 10)$)

Fig. 3 Decoding performance of Chase decoding algorithm with adaptive estimating SNR

表 2 不同算法对平均不可靠比特数目以及程序运行时间的影响 (迭代 5 次, 数据量为 9.61×10^8)

Table 2 Effects of different decoding algorithms on the average No of unreliable bits and program running time (9.61×10^8 Bits at iteration 5)

SNR/dB	平均不可靠比特数目			程序总运行时间/s		
	Chase - 2	自适应估计 SNR	每次迭代自适应估计 SNR	Chase - 2	自适应估计 SNR	每次迭代自适应估计 SNR
1.0	3	2.267 76	1.086 030	381.77	284.12	245.52
1.5	3	1.776 54	0.844 012	369.28	240.32	214.88
2.0	3	1.420 25	0.665 012	380.57	223.43	192.88
2.5	3	1.153 34	0.525 141	373.00	208.19	183.97
3.0	3	0.936 39	0.412 079	371.75	194.70	172.36

3.3 结果分析

自适应估计 SNR 的 Chase 算法与每次迭代自适应估计 SNR 的 Chase 算法具有以下优势:

(1) 能够降低译码复杂度。虽然自适应估计 SNR 对于每一个乘积码需要 n^2 次加法, n^2 次乘法, 但是相对于不可靠比特数目的减小引起译码复

3.2 每次迭代自适应估计 SNR 的 Chase 算法

为了进一步简化译码复杂度, 利用 SNR 估计值随迭代次数递增的关系, 在分组 Turbo 码的每一次迭代译码中应用自适应估计 SNR 的算法。自适应改变门限函数 $T(\gamma)$ 的大小, 可减少后几次迭代中的不可靠比特的数目。

表 2 示出了 $(31, 26, 3)^2$ BTC 译码中不同算法对平均不可靠比特数目以及程序运行时间的影响。不难发现, 相对于 Chase - 2 算法, 自适应估计 SNR 的 Chase 算法能够减小平均不可靠比特数。而每次迭代自适应估计 SNR 的 Chase 算法与自适应估计 SNR 的 Chase 算法相比, 能够进一步降低译码复杂度。同时, 对于程序总运行时间, 前者的运算时间也更短。可见, 每次迭代自适应估计 SNR 的 Chase 算法不仅没有因为多次估计 SNR 造成运算时间的增加, 相反, 更新的 SNR 降低了平均不可靠比特数, 缩短了程序运行时间。

每次迭代自适应估计 SNR 的 Chase 算法的译码性能如图 4 所示。可见, 该算法与 Chase - 2 算法相比, 不仅能够降低译码复杂度, 而且译码性能也无明显降低。前 3 次迭代, 每次迭代自适应估计 SNR 的 Chase 算法译码性能略逊于 Chase 算法; 而第 5 次迭代, 则优于 Chase 算法。在迭代 5 次, $E_b/N_0 = 3 \text{ dB}$ 时, 其 BER 可以达到 10^{-7} 量级。

杂度的减小还是非常值得的。当 $E_b/N_0 = 3 \text{ dB}$ 迭代 5 次时, 自适应估计 SNR 的 Chase 算法的平均不可靠比特数仅为 Chase - 2 算法的 30%, 硬判决译码次数为 Chase - 2 算法的 23%; 而每次迭代自适应估计 SNR 的 Chase 算法, 其平均不可靠比特数仅为 Chase - 2 算法的 13.7%, 硬判决译码次数

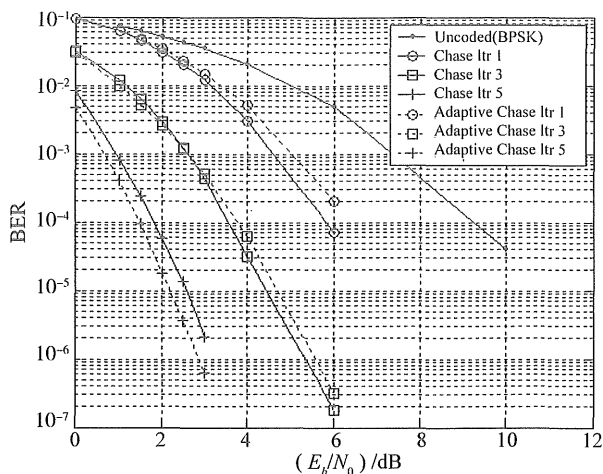


图4 每次迭代自适应估计 SNR 的 Chase 算法 (Adaptive Chase) 译码性能

Fig. 4 Performance of Chase decoding algorithm with adaptive estimating SNR in each iterative decoding

为 Chase-2 算法的 16.6%。

(2) 能够缩短译码时间。估计 SNR 带来的运算量并没有增加译码时间, 相反, 由于更新的 SNR 加快了译码时间, 提升了译码效率。实验结果表明, 当 $E_b/N_0 = 3$ dB 迭代 5 次时, 自适应和每次迭代自适应估计 SNR 的 Chase 算法程序运行总时间分别仅为 Chase-2 算法的 52.2% 和 46%。

(3) 能够一定程度提升译码性能。经多次迭代后自适应估计 SNR 的 Chase 算法不仅能降低译码复杂度, 还能提升译码性能。在迭代 5 次, $BER = 1.2 \times 10^{-5}$ 时, 自适应估计 SNR 的 Chase 算法相对于 Chase-2 算法可以获得 0.5 dB 的编码增益。

(4) 不需要预先设定信噪比的值, 可以自动估计信噪比, 还能针对信噪比的变化自动调整译码复杂度, 改善译码性能。

4 结 论

本文提出了两种新的分组 Turbo 码译码方案, 即自适应估计 SNR 的 Chase 算法和每次迭代自适应估计 SNR 的 Chase 算法, 并采用这两种算法分别对 $(31, 26, 3)^2$ BTC 码进行了实验。结果表明, 自适应估计 SNR 的方法能够避免在自适应 Chase 译码算法中使用预置值, 能够自适应调整门限函数, 改变不可靠比特数。同时, 选用适当的量化因子, 不仅能缩短运算时间, 还能够改善译码的 BER 性能。将自适应估计 SNR 的方法应用于每次迭代译码中, 能进一步降低译码复杂度。

参考文献:

- [1] BERROU C, CLAVIEUX A, THITIMAJSHIMA P. Near Shannon limit error-correcting coding and decoding: Turbo-codes[C]. ICC'93, Geneva, 1993:1064-1070.
- [2] BERROU C, CLAVIEUX A. Near optimum error correcting coding and decoding: Turbo-codes[J]. IEEE Trans on Comm, 1996, 44(10): 1261-1271.
- [3] PYNDIAH R, GLACIEUX A, PICART A, et al. Near optimum decoding of product codes [C]. GLOBECOM'94, San Francisco, CA, 1994:339-343.
- [4] PYNDIAH R. Near-optimum decoding of product codes: block Turbo codes[J]. IEEE Trans on Comm, 1998, 46(8): 1003-1010.
- [5] 张薇, 刘星成, 赵连强. 分组 Turbo 码软判决自适应 Chase 译码算法的研究[J]. 中山大学学报: 自然科学版, 2005, 44(4): 47-50.
ZHANG Wei, LIU Xingcheng, ZHAO Lianqiang. Study of adaptive chase algorithm for block codes[J]. Acta Scientiarum Naturalium Universitatis Sunyatseni, 2005, 44(4): 47-50.
- [6] ARMAND M A, HALIM A, NALLANATHAN A. Chase decoding of linear Z4 codes at low to moderate rates[J]. IEEE Comm Letters, 2007, 11(10): 811-813.
- [7] VICENTE C B, WEBER J H. Dynamic Chase decoding algorithm[C]. ITW'2003, Paris, 2003: 312-315.
- [8] ARICO G, WEBER J H. Limited-trial Chase decoding [J]. IEEE Trans on Info Theory, 2003, 49(11): 2972-2975.
- [9] WEBER J H. Low-complexity Chase-like bounded-distance decoding algorithms [C]. GLOBECOM'03, San Francisco, CA, 2003, 3: 1608-1612.
- [10] CHI Z, SONG L. On the performance/complexity tradeoff in block Turbo decoder design[J]. IEEE Trans on Comm, 2004, 52(2): 173-175.
- [11] MAHRAN A, BENAÏSSA M. Adaptive Chase algorithm for block Turbo codes[J]. Electronics Letters, 2003, 39(7): 617-619.
- [12] HANZO L, WOODARD J P, ROBERTSON P. Turbo decoding and detection for wireless applications [J]. Proc of the IEEE, 2007, 95(6): 1178-1200.
- [13] SUMMERS A, WILSON G. SNR Mismatch and online estimation in Turbo decoding [J]. IEEE Trans on Comm, 1998, 46(7): 421-423.
- [14] ZHOU R, BIDAN R L, PYNDIAH R, et al. Low-complexity high-rate Reed-Solomon block Turbo codes [J]. IEEE Trans on Comm, 2007, 55(9): 1656-1660.