

低复杂度 Chase-2 型级联码译码器

刘晓健, 赵春明, 孙小钧

东南大学移动通信国家重点实验室, 南京 210096

E-mail: liuxiaojian_cn@seu.edu.cn

2008-11-21 收稿, 2009-07-08 接受

国家高技术研究发展计划(编号: 2006AA01Z263)、东南大学移动通信国家重点实验室自主研究课题(编号: 2008A10)资助项目

摘要 由里德-所罗门码(RS 码)和递归系统卷积码(RSC)组成的级联码被广泛应用于现有的通信系统中, 如卫星通信系统和 WiMAX 系统. 介绍了一种用于这类级联码的低复杂度迭代软判决译码算法. 通过减少测试图样的使用数量, 对 chase-2 型算法进行了简化, 达到了减少运算复杂度的目的; 此外, 通过改进内外码解码器之间软信息的处理方法, 增加了软信息的使用效率, 改善了译码性能. 仿真结果表明, 改进型译码算法在性能和计算复杂度上取得了良好的平衡.

关键词

级联码
RS 码
RSC 码
Chase 译码算法
BCJR 算法

级联码方案最早由 Forney^[1]提出, 它一般由两种纠错码分别作为内码和外码级联构成, 其中由 RSC 码作为内码, RS 码作为外码的级联码方案被许多通信系统所采用, 如卫星通信系统^[2]和 WiMAX^[3]系统.

为了取得最优的译码性能, 级联码的译码器往往采用迭代软判决译码算法. RSC 码的软判决译码算法有 SOVA^[4] 和 BCJR^[5]算法; RS 码的则有 Chase 型译码算法^[6], GMD 算法^[7], K-V 算法^[8], 自适应置信度传递算法 (ABP)^[9]等, 其中 Chase-2 型算法在复杂度和性能上具有良好的平衡, 被大多数系统所采用. 它的另一个优点是, 不用将 RSC 译码器输出的比特级软信息转换成 RS 码的符号级软信息, 从而减少了转换过程中的信息损失, 同时也降低了计算复杂度.

对于一个码长为 n , 信息位长度为 k , 最小汉明距离为 d_{\min} 的二进制 (n, k, d_{\min}) 线性分组码, Chase-2 型译码算法要根据接收序列中的 $\lfloor d_{\min}/2 \rfloor$ 个最不可靠位置(LRPs), 加载 $2^{\lfloor d_{\min}/2 \rfloor}$ 个测试图样, 并

分别译码. d_{\min} 较大时, 算法的复杂度非常高, 因此在实际的应用中, 通常希望在保证性能的同时, 能够尽量降低这种算法的复杂度.

本文首先讨论了 Chase-2 型算法的简化问题. 简化的 Chase-2 型算法最多只使用 2^{ρ} ($\rho < (d_{\min}/2)$) 个测试图样, 同时使用了一种简单的译码中止条件. 然后, 文章介绍了软信息在内码和外码解码器之间的传递方法, 该方法能够有效地改善算法的性能. 仿真结果显示, 文章中提出的简化译码算法不仅计算复杂度低, 而且性能良好.

1 级联码方案

在本文介绍的级联码方案中, 内码为 RSC 码, 外码为 RS 码. 图 1 显示了该方案的编解码流程.

在发送端, 输入的数据流首先送往 RS 编码器. 编码后输出的码字经过交织被送往卷积码编码器. 在接收端, 收到的数据首先由 BCJR 解码器解码, 经过函数 1 和解交织处理后送往 RS 解码器. RS 解码器输出的软信息经交织和函数 2 的处理后, 被反馈给 BCJR 译码器, 这样就完成了一次迭代译码.

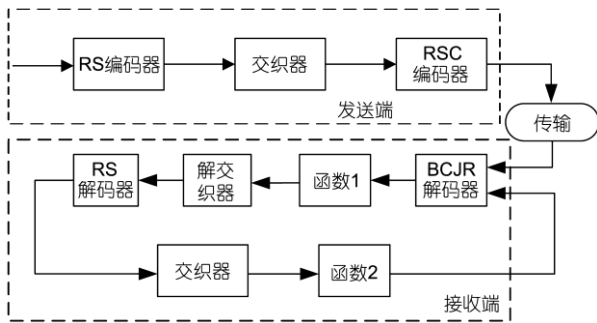


图1 级联码的编解码方案

交织和解交织技术是对抗突发错误(burst errors)的有效手段,它们通过将连续的突发错误分散到各个RS码字中去,增加了系统抵抗突发错误的能力^[10].本文中将要介绍的译码算法使用经典的符号级行列交织算法.一个深度为 I 的交织器把 I 个RS码字按照如下顺序排列成一个 $I \times n$ 的符号阵列(n 是每个RS码字的长度):

	符号阵列
RS码字 1	$s_1^1, s_2^1, \dots, s_n^1$
RS码字 2	$s_1^2, s_2^2, \dots, s_n^2$
⋮	⋮
RS码字 I	$s_1^I, s_2^I, \dots, s_n^I$

然后,阵列中的符号被按照如下顺序输出: $s_1^1, s_1^2, \dots, s_1^I, s_2^1, s_2^2, \dots, s_2^I, \dots, s_n^1, s_n^2, \dots, s_n^I$.

2 简化的 Chase-2 型算法

假设传输信道是加性高斯白噪声(AWGN)信道,符号采用BPSK调制,调制后的符号取值为 $\{-1, +1\}$.RS码解码器收到的序列 R 可以由下式表示:

$$R = C + N, \quad (1)$$

其中 C 是发送的码字, N 是方差为 σ^2 的加性高斯白噪声(AWGN).

Chase-2型算法执行步骤如下^[11]:

- (i) 通过对 R 进行硬判决得到 $\{0, 1\}$ 序列 Z ,给出 C 中每个符号的可靠度.
- (ii) 每次从 E 中选取一个测试图样 e ,得到修正的序列 $Z + e$.
- (iii) 用代数解码器对 $Z + e$ 进行一次代数译码得到候选码字 v .
- (iv) 对每个候选码字计算软判决信息,按照一定的准则从中选取解码结果.

集合 E 称为测试图样集,它包含 $2^{\lfloor d_{\min}/2 \rfloor}$ 个测试图样.这些图样通过在 Z 的 $\lfloor d_{\min}/2 \rfloor$ 个LRPs上加上所有可能的0,1组合得到.因此,测试图样集的大小和 d_{\min} 成指数关系.通常,为了减少计算复杂度只选取 ρ ($\rho < d_{\min}/2$)个LRPs.假设 $\rho = 2$, E 中的测试图样按照以下顺序给出:

00,01,10,11,

其中每个图样左边第一个比特代表最不可靠位置,右边的比特则代表次最不可靠位置.

简化的 Chase-2 型算法的具体的步骤如下所示:

- (i) 通过对 R 进行硬判决得到序列 Z ,给出 C 中每个符号的可靠度.
- (ii) 每次从 E 中选取一个测试图样 e ,得到修正的序列 $Z + e$.
- (iii) 用代数解码器对 $Z + e$ 进行一次代数译码得到候选码字 v .
- (iv) 计算 v 的校正子,如果所有校正子都为0,则终止译码并将 v 作为结果输出.否则,重复步骤(ii),(iii)直至满足校验的码字被找到,或者 E 中的测试图样都被用完.

简化的 Chase-2 算法仅仅选取第一个通过校验的码字作为译码器的输出,而不是测试所有的测试图样,因此减少了运算量.简化算法的译码性能非常接近于原算法,这是因为:

- (i) 对大多数这类RS码字,不考虑测试图样,常常一次代数译码就可以得到正确的码字.图2显示了测试图样和译码正确概率之间的统计关系,其中的数据是通过RS(63,49,15)进行仿真得到的($\rho = 4$),仿真时的误码率(BER)约为 10^{-5} .

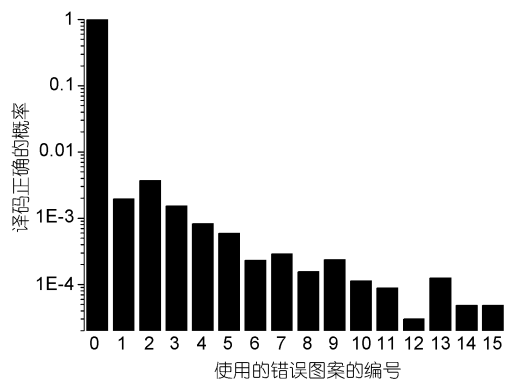


图2 测试图样的使用和正确译码概率的统计关系

(ii) RS 码的 d_{\min} 较大时, Chase-2 型算法通过多个不同的测试图样常常得到相同的译码结果. 这种情况下, 选择第一个通过校验的译码码字作为输出导致的错误概率比较低.

为了进一步说明简化算法的有效性, 对 RS(63, 49, 15)进行了仿真结果如图 3 所示. 由图可见, 简化对算法的性能影响不明显. 图中的信噪比由 $10 \cdot \lg\left(\frac{1}{2t \cdot \sigma^2}\right)$ 计算得到, 其中 t 表示码率.

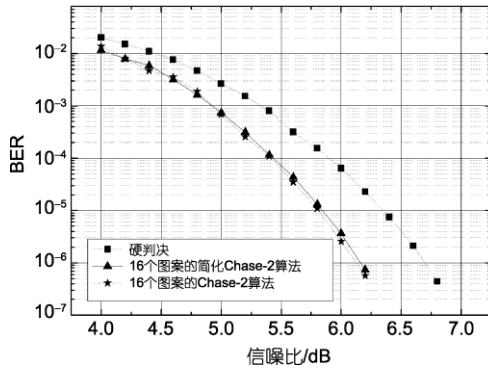


图 3 各种 RS 解码算法的性能比较

3 级联码系统的解码

3.1 BCJR 解码器

Viterbi 算法在各个比特的错误概率均等的情况下表现良好, 然而此条件不满足时, 使用 BCJR 算法能取得更好的性能. 在迭代译码中, 因为信息比特的先验概率随着迭代的进行不断改变, 所以通常使用 BCJR 算法.

大多数情况下, 内码是码率为 0.5 的 RSC 码. 假设在时刻 l 编码器接收的比特为 u_l , 编码后得到一对二进制数 $(x_l; y_l)$, 其中 $x_l = u_l$.

在 AWGN 信道和采用 BPSK 调制的条件下, 解码器在时刻 l 的输入是一对变量 rx_l 和 ry_l .

$$\begin{cases} rx_l = x_l + n_{1l}, \\ ry_l = y_l + n_{2l}, \end{cases} \quad (2)$$

其中 n_{1l} 和 n_{2l} 是两个独立的噪声采样, 他们的方差都为 σ^2 .

解码器的输入除了 rx_l 和 ry_l 外, 还有信息序列的先验信息 $L_a(u_l)$, $l=0,1,\dots,h-1$, 其中 h 是信息序列的长度. 在迭代译码系统中, $L_a(u_l)$ 通过对外码译码结果的软判决得到.

BCJR 解码器的输出是比特序列的后验概率信息 (APP L 值):

$$L(u_l) = \ln \left[\frac{p(u_l = +1|r)}{p(u_l = -1|r)} \right], \quad (3)$$

其中 r 表示由 rx_l 和 ry_l 组成的序列.

3.2 内外码解码器之间软信息的处理方法

输入 RS 解码器的是解交织后的 BCJR 译码器的输出. 在对 RS 码译码之前, 首先要对这些 APP L 值进行如下修正:

$$L'(u_l) = \alpha \cdot L(u_l), \quad (4)$$

其中 α 是一个随着迭代次数变化的值. 接着, 用下面的式子进一步修正:

$$L''(u_l) = \begin{cases} -\beta \cdot L'(u_l), & \text{if } L'(u_l) \cdot rx_l < 0 \quad a < |L'(u_l)| < b, \\ L'(u_l), & \text{其他,} \end{cases} \quad (5)$$

式中 a 和 b 是小于 1 的常数, β 是大于 1 的常数.

$L'(u_l) \cdot x_l < 0$ 时, 有两种可能: (1) rx_l 被噪声污染, 但被 BCJR 解码器纠正; (2) BCJR 译码器给信息比特 u_l 提供了错误的 APP L 值. 通过适当的选择 a 和 b 可以尽量地对第二种情况中的错误进行纠正. 当 $L'(u_l)$ 经过(5)式中的第一个式子修正后, 可以认为一个错误被纠正, 它对应的位置应当从 RS 解码器的 LRP 列表中除去, 因此 β 被设为大于 1 的常数.

假设 RS 码是基于 $GF(2^m)$ 的, I 个 RS 码解码器输出的符号序列要被映射成如下的长度为 $n \cdot I \cdot m$ 的 ± 1 序列:

$$D = \underbrace{d_1, d_2, \dots, d_{n \cdot m}}_{\text{codeword 1}}, \underbrace{d_{n \cdot m+1}, d_{n \cdot m+2}, \dots, d_{(n+1) \cdot m}}_{\text{codeword 2}}, \dots,$$

该序列通过两种方法被反馈给 BCJR 解码器:

(1) 计算先验概率信息,

$$L_a(u_l) = \gamma \cdot d_l, \quad l=0,1,\dots,n \cdot I, \quad (6)$$

其中 $\gamma (\gamma < 1)$ 是一个随着迭代次数变化的值.

(2) 如果 d_l 来自一个通过校验的 RS 码字, 用它来修正 rx_l :

$$rx_l = d_l \cdot |rx_l|, \quad l=0,1,\dots,n \cdot I. \quad (7)$$

(5)和(7)等效于在 RS 码和 RSC 码上各加上了一个测试图样. 由于这些图样是基于软判决信息得到的, 比较准确, 所以能较为明显地改善译码器性能. 图 4 显示了上述的软信息处理过程.

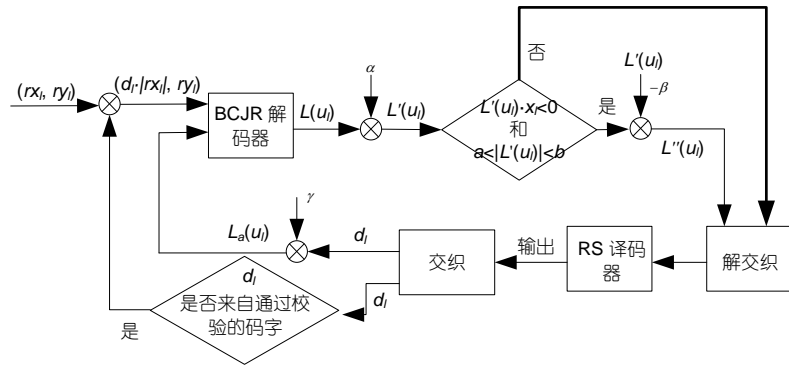


图4 软信息在接收端的处理流程

4 仿真结果

本节比较了以下 4 种算法的译码性能：(1)BM/BCJR 算法，内外码解码器之间使用文献[10]中的软信息处理方式；(2)Chase-2/BCJR 算法，内外码解码器之间使用文献[10]中的软信息处理方式；(3)简化的 Chase-2/BCJR 算法，内外码解码器之间使用文献 [10] 中的软信息处理方式；(4)简化的 Chase-2/BCJR 算法，内外码解码器之间使用 3.2 节给出的软信息处理方式。算法(1)中的 BM 算法是一种经典的代数译码算法。算法(2)由文献[10]中给出，这里用 BCJR 来代替原文中的 SOVA 算法以方便对比各种算法的性能。 ρ 在算法(2)中设为 4；在算法(3)和(4)中设为 2。

仿真采用 AWGN 信道，调制方式为 BPSK。1/2 码率的 RSC 码作为内码，生成多项式为 $G1 = 23, G2 = 35$ 。交织器深度 I 为 6，迭代译码的次数设为 4，参数 β 设为 2， a 和 b 分别设为 1.8 和 3.5。 α 和 γ 的设定如表 1 所示。图 5 显示了上述 4 种算法在 RS (63, 47,

17)作为外码时的译码性能。

由上图可见，算法(4)的性能最好，在 BER 为 10^{-5} 时，相比算法(2)有约 0.2 dB 的增益。算法(2)和(3)的性能非常接近。算法(4)的性能增益主要来自本文提出的内外码解码器之间的软信息传递方法。

表 1 各次迭代中 α 和 γ 的设定

迭代次数	1	2	3	4
α	0.6	0.8	1	1
γ	0.4	0.7	0.9	0.9

RS(255, 223, 33)作为外码时，各种算法的性能比较结果如图 6 所示。算法的各项参数设置和之前相同。和图 5 中的情况类似，算法(4)的性能最好，在 BER 为 10^{-5} 时，相比算法(2)有约 0.1 dB 的增益。算法(3)的性能依然介于算法(1)和(2)之间。

通过以上两个例子，可以得出以下结论：当外

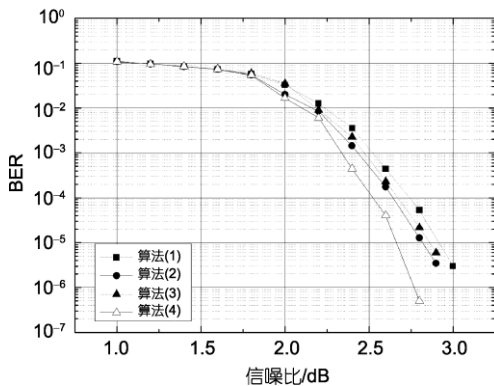


图5 各种算法在 RS (63, 471, 17)作为外码时的译码性能

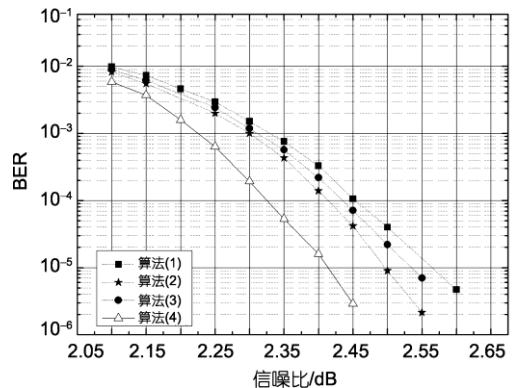


图6 各种算法在 RS (255, 223, 33)作为外码时的译码性能

码解码器使用简化的 Chase-2 型算法时, 性能的损失不明显. 在此基础上使用本文提出的内外码解码器之间的软信息处理方法后, 可以取得更好的性能.

5 结论

为了解决使用迭代软判决译码算法对由里德-

所罗门码(RS 码)和递归系统卷积码(RSC)组成的级联码进行解码时, 计算复杂度比较高的问题, 对 Chase-2 型算法进行了简化; 此外, 通过改进内外码解码器之间软信息的处理方法, 增加了软信息的使用效率, 改善了译码性能. 仿真结果表明, 改进型译码算法在性能和计算复杂度上取得了良好的平衡.

参考文献

- 1 Forney G D. Concatenated Codes. Cambridge: MIT Press, 1966
- 2 Digital broadcasting systems for television, sound and data services; Framing structure, channel coding and modulation for 11/12 GHz satellite service. ETS 300 421, 1994
- 3 IEEE standard for local and metropolitan area networks Part 16: Air interface for fixed broadband wireless access systems. IEEE Std 802:16TM-2004, 2004
- 4 Hagenauer J, Hoehner P, Viterbi A. Decoding algorithm with soft-decision outputs and its application. In: Proc IEEE Global Conference on Communications Dallas, Tex, 1989, 1680—1686
- 5 Bahl L R, Cocke J, Jelinek F, et al. Optimal decoding of linear codes for minimizing symbol error rate. IEEE Trans Inform Theory, 1974, 20: 284—287[doi]
- 6 Chase D. A class of algorithms for decoding block codes with channel measurement information. IEEE Trans Inform Theory, 1972, 18: 170—182[doi]
- 7 Forney G D Jr. Generalized minimum distance decoding. IEEE Trans Inform Theory, 1966, 1: 125—131[doi]
- 8 Koetter R, Vardy A. Algebraic soft-decision decoding of Reed-Solomon codes. IEEE Trans Inform Theory, 2003, 49: 2809—2825[doi]
- 9 Jing J, Narayana N. Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity check matrix. IEEE Trans Inform Theory, 2006, 52: 3746—3756[doi]
- 10 Aitsab O, Pyndiah R. Performance of concatenated Reed-Solomon/convolutional codes with iterative decoding. IEEE Globe Com, 1997, 2: 934—938
- 11 Shu L, Daniel J C Jr. Error Control Coding. 2nd ed. New York: Prentice Hall, 1983