

Turbo 码编译码原理及其性能分析

喻文芳, 周 辉

(装备指挥技术学院 电子工程系, 北京 101416)

摘 要: 讨论了 Turbo 码的编译码基本原理, 对 Turbo 码的性能进行了分析; 编码采用并行级联的循环系统卷积码, 译码器则为迭代的串行级联译码器, 进行随机交织, 经过高斯白噪声信道传输; 进行了计算机仿真, 比较了对数的最大似然译码算法与软输出的 Viterbi 译码算法在不同信噪比条件下的比特误码率。

关 键 词: 循环系统卷积码; Turbo 码; 软判决 Viterbi 译码; 交织器; 比特误码率

中图分类号: TP 911.22

文献标识码: A

文章编号: CN 11-3987(2003)03-0057-04

在数字通信系统中, 信道编码常用来保护系统免遭噪声和外界干扰, 并用于降低系统的比特误码率, 提高系统的可靠性。

近年来通信技术发展日新月异, 使得对信息传输质量的要求越来越高, 对信道编码的要求也越来越高。1993 年 Turbo 码的提出^[1], 为信道编码注入了新的活力。Turbo 码的性能以接近 Shannon 编码理论的极限而备受关注。近年来, 国内也开始重视 Turbo 码的理论研究和应用研究^[2,3]。本文就 Turbo 码的编译码原理进行讨论, 并通过计算机仿真分析了其优越的性能。

1 Turbo 码编译码原理

Turbo 码由 2 个循环系统卷积码 (recurrence system code, RSC) 并行级联而成: 译码采用迭代的串行译码; 交织器是 Turbo 码所特有的, 它可以使得信息序列随机化, 增加各码字间的重量, 从而提高码的保护能力。下面就 Turbo 码编码原理、交织器的选择和译码原理进行讨论。

1.1 Turbo 码编码原理

典型的 Turbo 码编码器由 2 个相同的 RSC 编码器和交织器等构成^[2], 其原理如图 1 所示。编码器 I、II 又称为成员码, 都为非循环非系统卷积码编码器, 共同构成了 RSC 编码器。编码器 I、II 只输出编码的校验序列, 通过开关单元 (有

必要, 则进行增信删余) 与输入信息序列一起输出得到 Turbo 码编码输出序列。例如, 对于生成矩阵为 $g=[g_1, g_2]$ 的 $(2, 1, 2)$ 卷积码通过编码后, 如果进行增信删余, 得到码率为 $1/2$ 的编码输出序列; 如果不进行增信删余, 则得到码率为 $1/3$ 的序列。

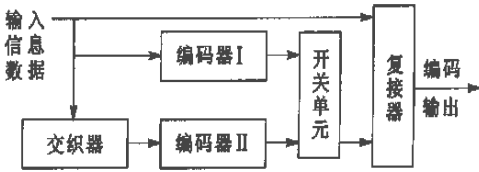


图 1 Turbo 码编码原理图

Turbo 码之所以利用 RSC 编码器, 是因为循环编码器可以改善码的比特误码率性能。从状态转移可以看出, 对于生成矩阵为 $g=(07, 05)$ 的 $(2, 1, 2)$ 非循环卷积码编码器状态转移如图 2(a) 所示; 经循环编码后, 其状态图变为图 2(b)。2 个状态转移图非常相似, 它们具有相同的生成矩阵、最小自由距离和网格结构; 但是, 循环编码后输出码字的重量增加了, 从而提高了比特误码率性能。

对于卷积码编码器, 通过在输入信息序列后加入 m (编码约束长度) 个“0”, 就可以使编码状态回到全“0”状态, 使网格终止; 但对于 RSC 编码器, 由于有反馈, RSC 编码器的状态很难确定, 加“0”的策略不能达到网格终止的目的; 而且, 在编

收稿日期: 2002-10-08
作者简介: 喻文芳(1972—), 女(汉族), 四川眉山人, 硕士, 讲师。

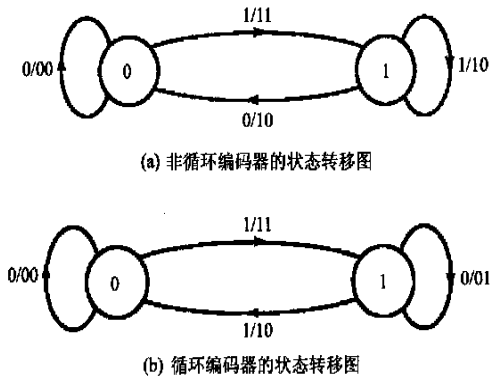


图2 卷积码状态结构图

码器 I 回到了全“0”状态,由于加入了交织器,编码器 II 却不一定回到全“0”状态,所以需要寻求别的办法来终止网格。最简单的方法就是在输入端设置开关电路来实现网格的终止^[3]。方法之二为:初始状态为全“0”,在 RSC 编码器 I 的信息序列后加入 m 个“0”比特,使其状态回到全“0”,从而使网格终止;而编码器 II 不终止,处于开放状态,可以在任何状态停止编码。

1.2 交织器的选择

Turbo 码有很好的性能,原因之一就在于交织器的使用。交织器的选择对 Turbo 码的整体性能能有非常大的影响。

交织器实际上是一个置换器,其设计应满足:最大程度地置乱原数据排列顺序和避免置换前相邻 2 个数据在置换后仍然相邻,即把低重量的输入序列中连续“1”的位置拉开,并使编码后的码字具有高重量。

交织器有多种,如随机交织器、块交织器、循环移位交织器和奇偶交织器等^[4]。各种交织器有各自的特点。

块交织器在通信系统中应用比较多,它简单、有效。其基本原理为:列写入,自上而下,从左到右;行读出,从左到右,自上而下。如图 3 所示。

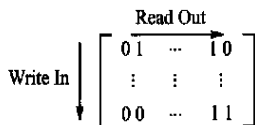


图3 块交织器

随机交织器则利用固定的随机置换方法,将输入序列映射为置换规则的顺序。

循环移位交织器的置换规则 p 被定义为

$$p(i) = (ai + s) \bmod L$$

其中: a 为步长, $a < L$; s 为偏移值, $s < L$; i 为索引值。如 $L = 8, a = 3, s = 0$,如果交织器写入 $[0\ 1\ 1\ 0\ 1\ 0\ 0\ 1]$,则经过循环移位交织后读出为 $[0\ 0\ 0\ 1\ 1\ 1\ 1\ 0]$ 。可以看出,分开的相邻比特

位置为 3 或 5。

循环移位交织器将低码字重量为 2 的输入序列,置换为具有高码字重量的输入序列非常有效。但是,由于该置换方法本身的规律性,很难将具有低码字重量大于 2 的输入序列置换为具有高码字重量的其他输入序列。

奇偶校验交织器专为码率为 1/2 的 Turbo 码设计的,用以克服增信删余的不足。如图 1 所示的编码原理,如果编码后长度为 $L = 9$ 的信息序列 $X = \{x_1, x_2, \dots, x_9\}$,编码器 I 输出的校验序列 $C_2 = \{c_{21}, c_{22}, c_{23}, \dots, c_{29}\}$,编码器 II 输出的校验序列 $C_3 = \{c_{31}, c_{32}, c_{33}, \dots, c_{39}\}$,对信息序列 X 使用 3×3 的块交织器进行交织,则 X 按列写入为 $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$,再按行读出为 $\{x_1, x_4, x_7, x_2, x_5, x_8, x_3, x_6, x_9\}$ 。奇偶校验交织器则选择 C_2 的奇数位、 C_3 的偶数位与信息比特相对应。对于码率为 1/2,编码序列经复用后的输出序列为 2×9 的矩阵

$$\begin{bmatrix} x_1 & x_4 & x_7 & x_2 & x_5 & x_8 & x_3 & x_6 & x_9 \\ c_{21} & c_{34} & c_{27} & c_{32} & c_{25} & c_{38} & c_{23} & c_{36} & c_{29} \end{bmatrix}$$

在实际应用中,则可根据实际情况选用合适的交织器。

1.3 Turbo 码译码原理

卷积码的 Viterbi 译码算法可以得到最大似然输出序列,是一种最优的算法。但对于级联的卷积码,传统的 Viterbi 译码器有 2 大缺陷^[5]:内部的 Viterbi 译码器产生的突发错误会降低外部的 Viterbi 译码器的性能;内部的 Viterbi 译码器的硬判决输出,使得外部的 Viterbi 译码器不能利用软判决带来的好处,但引入可靠性参数就可以提高级联译码系统的性能。如果 Viterbi 译码器能产生可靠性参数(软输出),再将该软判决输出值输入到随后的 Viterbi 译码器中作为先验概率来提高译码性能。这种修正的 Viterbi 译码算法构成的译码器称为软输出的 Viterbi 算法(soft output Viterbi arithmetic, SOVA)译码器^[6],图 4 所示为级联的 SOVA 译码器框图。

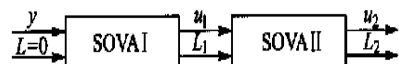


图4 级联的 SOVA 译码器

其中: L 为相关的可靠性值,与信道参数有关,可以定义为信噪比的函数^[6], $L = 4 E_b / N_0$ (E_b 为单位信号能量, N_0 为单边带噪声功率); y 为信道接收值; u 为硬判决输出。

Turbo 码译码算法有多种:最大似然译码 MAP、对数的最大似然译码(Log-MAP)、基于修正的 Viterbi 译码算法等。从性能好坏和实现难

易程度来看, MAP 译码精度高, 但延迟大, 存储量大, 不易实现; 而软判决算法, 如 SOVA 算法, 通过引入可靠性参数(软输出)可以改善译码性能, 同时译码延迟和计算复杂度低, 适合于硬件实现, 所以比较常用。图 5 表示 Turbo 码迭代的 SOVA 译码原理图, 图中: I 为交织器; I^{-1} 为解交织器; C_{1k}, C_{2k} 为校验序列; X_k 为信息序列; L_{1k}, L_{2k} 为可靠性值, 即 SOVA 译码器的软输出值; Y_{1k} 为经 SOVA I 译码的估值; Y_k 为译码得到的估值。

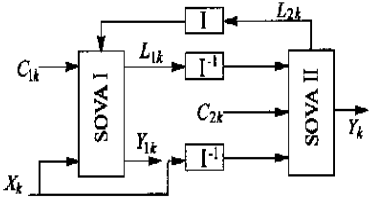


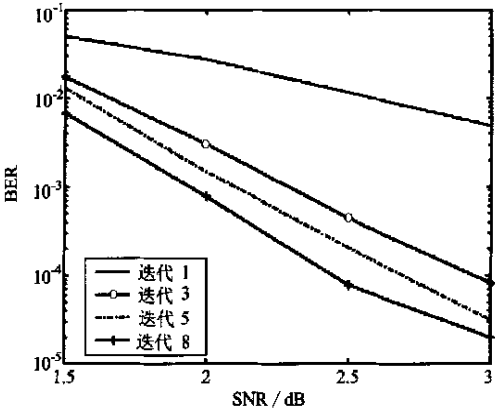
图 5 turbo 码迭代的 SOVA 译码原理图

2 仿真分析

在不同信噪比条件下, 利用 Matlab 语言对 Turbo 码的比特误码率进行仿真分析。编码部分: 用 2 个 RSC 编码器。译码部分: 用迭归的 2 个串行级联译码器, 译码算法分别采用 Log-MAP 和 SOVA 算法。采用随机交织器, 它代表了所有交织器的平均性能。编码后的 $\{0/1\}$ 序列进行 BPSK 调制为 $\{-1/+1\}$ 序列。信道选用高斯白噪声模型。

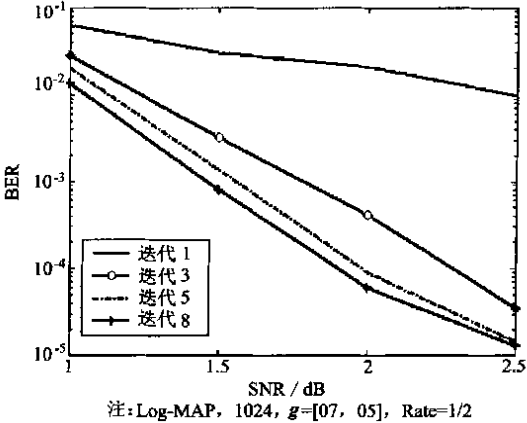
输入编码器的信息序列为均匀分布的随机序列, 原码是生成矩阵为 (07, 05) 的 (2, 1, 2) 卷积码; 进行增信删余后, 得到码率为 1/2 的 Turbo 码; 选择每一帧信号大小为 1 024, 迭代译码最高次数为 8 次, 交织器尺寸为 64×64 , 其信噪比 (SNR) 与比特误码率 (BER) 性能如图 6、图 7 所示。

从仿真结果可知: ① 在相同信噪比条件下, 迭代次数越多, 比特误码率越低; 但 Log-MAP 算法比 SOVA 算法要低 1 个数量级, 即有 0.5 dB 左右的增益。② 在迭代次数相同的情况下, 信噪比越高, 则误比特率越低。③ 从译码器实现上讲, 最大似然译码算法最复杂, 译码延迟大, 需要的存储量也大, 很难满足实际应用的需要; SOVA 算法实现复杂度较低, 易于硬件实现, 文献[1] 指出, 在迭代 18 次以上, 信噪比大于 0.7 dB 时, 译码比特误码率可达到 10^{-5} 。



注: SOVA 1024 $g=[07, 05]$, Rate=1/2

图 6 不同信噪比条件下 SOVA 算法的 BER 特性



注: Log-MAP, 1024, $g=[07, 05]$, Rate=1/2

图 7 不同信噪比条件下对数最大似然译码算法的 BER 特性

3 结 束 语

Turbo 码作为一种性能优越的信道码, 其应用逐步推广到深空通信、卫星通信和移动通信, 并被推荐为第三代移动通信的信道编码方案之一。但由于所有性能分析都建立在计算机模拟的基础上, 所以对 Turbo 码今后还有许多研究工作可做, 如 2 个串行译码器之间信息的传递方式、交织方法、如何利用 DSP 芯片实现 Turbo 码的编译码器等等。目前, 正进一步研究 Turbo 的实现问题, 将 Turbo 与调制技术相结合, 实现高速的调制解调器, 最终应用于卫星通信中。

参考文献:

[1] Berrou C, Glavieux A, Thitimajshima P. Near shannon limit error-correcting coding and decoding: Turbo Codes[A] . Proceeding of IEEE ICC' 93[C] , Geneva, Switzerland. 1064-1070.

[2] 赵雅兴, 张 宁. 改进的 Turbo 码算法的 FPGA 实现[J] . 信号处理, 2002, (6): 237-240.

[3] 徐澄圻, 陈晶沪. 一种基于SOVA改进 Turbo 码的译码方案 [J]. 通信学报, 1999, (12): 1-8.

[4] Stewart C. Turbo-code design issues: Trellis termination methods interleaving strategies and implementation complexity [A]. Proceeding of IEEE ICC' 99[C], Ottawa, Canada. 768-784.

[5] Joachim H, Elke O, Lutz P. Iterative decoding of binary block and convolutional codes[J]. IEEE Trans. Inform. Theory, 1996, 42(2): 429-445.

[6] Patrick R, Peter H. Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding[J]. European Trans. on Telecommun, 1997, 8(2): 119-125.

The Coding and Decoding Principles and Performance

Analysis of Turbo Codes

YU Wen-fang, ZHOU Hui

(Department of Electronic Engineering, the Academy of Equipment Command &Technology, Beijing 101416, China)

Abstract: The paper analyzes the performance of the turbo codes on the basis of the principles of the coding and decoding. The turbo code encoder is built using two identical recursive systematic convolutional (RSC) codes with parallel concatenation. The decoder is built using two iterative soft-output Viterbi algorithm decoders serially. We use random interleaver and an AWGN channel. The computer simulation compares the performance of bit error rate of the log-MAP and SOVA under the different signal-noise ratio.

Key words: recurrence system code; turbo codes; soft-out viterbi arithmetic; interleaver; bit error rate

(责任编辑:程万鑫)