

基于 Log-MAP 算法的 Turbo 码量化分析

闫涛, 杜兴民

(空军工程大学工程学院, 陕西西安 710038)

摘要: 针对 AGW 信道, 基于 Log-MAP 译码算法, 系统分析与研究了软输入软输出 (SISO) Turbo 迭代译码过程中数据的量化与定点实现。结合数据观察与算法剖析, 借助蒙特卡罗仿真给出了接收信息、外信息、信道可信度值、状态度量等数据的定点表示及量化细节, 所选择的量化方案达到了译码性能与实现复杂度较好的折衷, 量化后性能损失不大于 0.15 dB。

关键词: Log-MAP 算法; 量化; 接收信息; 信道可信度; 外信息; 状态度量

中图分类号: TN919 文献标识码: A 文章编号: 1009-3516(2005)04-0052-03

量化是工程开发的前提。Turbo 译码为软输出软输入迭代译码, 因此量化显得非常重要, 量化级数太小会造成较大的性能损失, 太大则会增加电路的复杂度及存储量, 事实上 Turbo 迭代译码过程中需要很大的存储量。我们选择量化方案要考虑到性能与复杂度较好的折衷。对于 Turbo 码的量化问题, 很多文献只是粗略的提及, 较详细的也只是讨论了接收信息、外信息与状态度量值的量化^[1~2], 许多细节并未给出。我们旨在从系统的角度对 Turbo 译码的量化进行分析研究, 通过 BER 性能仿真选择较好的量化方案。文献[3]对 Turbo 码的量化方式与量化级数进行了深入研究, 接收信息使用均匀量化方式, 而中间变量采取非均匀量化以提高译码性能, 但是非均匀量化增加了硬件实现的复杂度, 而且还会带来一定的译码延迟。这里所有的量化均采用均匀量化方式。

1 Turbo 码的 Log-MAP 译码算法与接收信息的量化

Log-MAP 译码算法是 Turbo 码最大后验概率 (MAP) 算法的对数域简化, 适于硬件实现, 它避免了 MAP 算法中大量的指数与对数运算, 却能达到与 MAP 算法几乎相同的译码性能^[4]。

定义函数 $\text{Max}(x, y) = \max(x, y) + \ln(1 + e^{-|x-y|})$, AWGN 信道下, Log-MAP 译码算法的对数似然比为 $L(a_k) = \text{Max}_{(m', m)}(\alpha_{k-1}(m') + \gamma_k(+1, m', m) + \beta_k(m)) - \text{Max}_{(m', m)}(\alpha_{k-1}(m') + \gamma_k(-1, m', m) + \beta_k(m))$ 。 m', m 分别对应 $k-1$ 和 k 时刻的编码器状态; α_k, β_k 分别称为前向和后向状态度量, 可由路径度量 $\gamma_k(+1, m', m)$ ($i = \pm 1$) 递推计算 $\alpha_k(m) = \text{Max}_{(m', m)}(\alpha_{k-1}(m') + \gamma_k(i, m', m); \beta_{k-1}(m') = \text{Max}(\beta_k(m) + \gamma_k(i, m', m))$ 。

外信息若采用 Robertson 使用方式^[5], 码率为 1/2 的 RSC 码路径度量 $\gamma_k(+1, m', m)$ 计算公式可以表示为 $\gamma_k(i, m', m) = \frac{1}{2} \cdot L_c \cdot i \cdot x_k + \frac{1}{2} \cdot L_c \cdot j \cdot y_k + \frac{1}{2} \cdot i \cdot z_k$ 。式中: $j = \pm 1$, 表示对应信息比特 $a_k = i$ 编码应输出的双极性校验比特; $L_c = 4E_s/N_0$ 定义为信道可信度值; x_k, y_k 为解调器输出的受噪声污染的信息比特与校验比特, 即译码器的接收信息; z_k 表示从另外一个成员译码器得到的外信息。

由于 Turbo 编译码是以块 (block) 为单位进行的, 因此对于接收到的信息要进行缓存, 缓存的大小直接影响器件的资源占用。选择较大的量化级带来的缓存以及电路复杂度的增加是不能容忍的, 而仿真表明, 选择过小的量化级则致使译码性能损失很大。综合数据的观察, 分别考虑 8、16、32 级量化。设量化的表示方

收稿日期: 2004-09-10

基金项目: 武器装备预研基金项目资助 (51473090104JB3201)

作者简介: 闫涛 (1979-), 男, 河南民权人, 硕士生, 主要从事保密通信与抗干扰技术研究;
杜兴民 (1941-), 男, 安徽涡阳人, 教授, 主要从事信息安全与对抗研究。

法为 $m:n$, 其中 m 表示总共需要的比特数, 而 n 表示小数部分需要的比特数, 以下均以这种方法描述。选取 3:1、4:1、4:2、5:2 和 5:3 量化与未量化的 Turbo 码译码性能比较, 如图 1, 仿真中采用的 RSC 码为(13, 15), 使用随机交织器, Bpsk 调制解调方式。所选信息帧长为 1 024 bit, 码率为 1/2。

由图 1 可以看出, 随着量化级的增加, 接收信息量化后的 Turbo 码译码性能逐步接近未进行量化时的性能。考察选取的几种量化方案, 对接收信息小数部分舍弃较大的 3:1、4:1 方案带来的性能损失相对较大, 这说明受噪声污染的接收信息小数部分对译码提供的信息量是不可低估的, 而从仿真结果来看, 过多的保留接收信息的小数部分所带来的性能提高并不明显, 如 5:3 并不比 4:2 方案的性能有太大优越性。我们选取 16 级的 4:2 量化方案, 该量化方案下的译码性能略逊于 5:3 方案, 对于信息帧长为 1 024 的情况下却可以节省 $2 \times 1 \times 1\,024$ bit 的缓存量, 同时可以减少后续计算电路的复杂度。

2 信道可信度值的定点表示与外信息的量化处理

在 Turbo 译码过程中需要对信道进行实时估计, 结合 Turbo 码的码率 R 通常对信道可信度值修正为 $L_c = 4_r/N_0$ 。信道信噪比失配对于 Turbo 码的 BER 性能具有一定的影响, 因此在定点实现中, 对信道可信度值的处理也是一个不可忽略的因素。 L_c 为信噪比的函数, 理论上 L_c 在一个无限大的正数范围内变化, 若把所有可能的值都定点表示是不现实的。这里我们将 L_c 固定在低信噪比(2.5 dB 以下)范围内, 这是因为一方面信噪比失配本身具有一个可以容忍的范围^[6], 另一方面, L_c 在路径度量计算中只是一个权值, SNR 失配仅造成迭代译码中外信息相对于接收信息的使用权重过大或过小, 在大信噪比条件下 Turbo 译码一般不进行迭代即不使用外信息即可完成译码, 故这样处理几乎不影响高信噪比下的译码性能。我们考察 1/2 码率情况下部分信道可信度值的范围, 采用 5:2 量化方式即共有 15 个定点值(除去 0)表示信道可信度值 L_c , 根据信道状况估计情况进行选择。

外信息用作成员译码器的先验信息, 其数值分布在很大的范围内, 大信噪比条件下其绝对值最大可达 30 以上, 若完全将所有外信息数值定点表示, 交织或解交织时带来的存储量是很大的, 外信息与译码比特先验概率的关系为 $P_r(a_k = +1) \approx e^{a_k} / (1 + e^{a_k})$; $P_r(a_k = -1) \approx 1 / (1 + e^{a_k})$ 。表 1 可以看出, 当表示外信息整数部分的比特数超过 4 时, 外信息所能提供给成员译码器的最大先验概率已近似为 1, 这说明此时量化后的外信息足以为成员译码器提供比较完整的译码参考信息。

表 1 几种外信息量化方案下的最大先验概率

外信息整数部分所用比特数	2	3	4	5	6	7
最大先验概率	0.731 058 58	0.952 574 13	0.999 088 95	0.999 999 69	1.000 000 00	1.000 000 00

图 2 可以看出, 在所考察的几种外信息量化方案中, 5:2 量化方案性能损失最大, 量化损失增益在大信噪比条件下比较明显, 其他几种量化方案的性能损失相对较小, 且 BER 性能很相近, 仿真结果可以用表 1 的数据进行解释, 当外信息的值足够覆盖整个先验概率空间时, 过多的比特表示外信息已经没有必要, 因此 6:2、7:2 与 7:1 量化方案下的 Turbo 码性能相差并不大, 综合考虑, 我们选择 6:2 量化方案。

接收信息、信道可信度值和外信息定点量化后, 路径度量即实现了定点表示, 仿真结果表明保留 2 比特

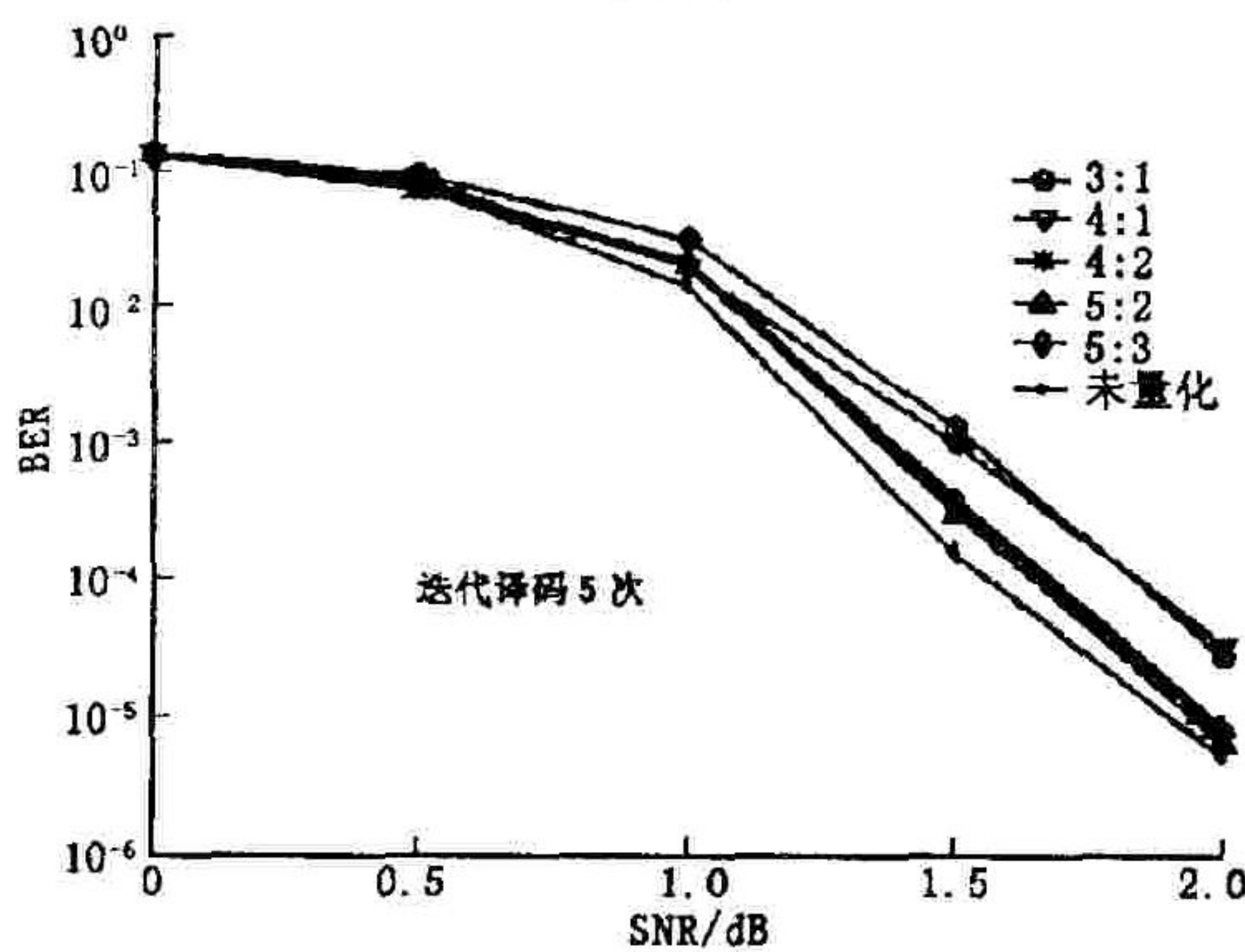


图 1 接收信息量化对 Turbo 码性能的影响

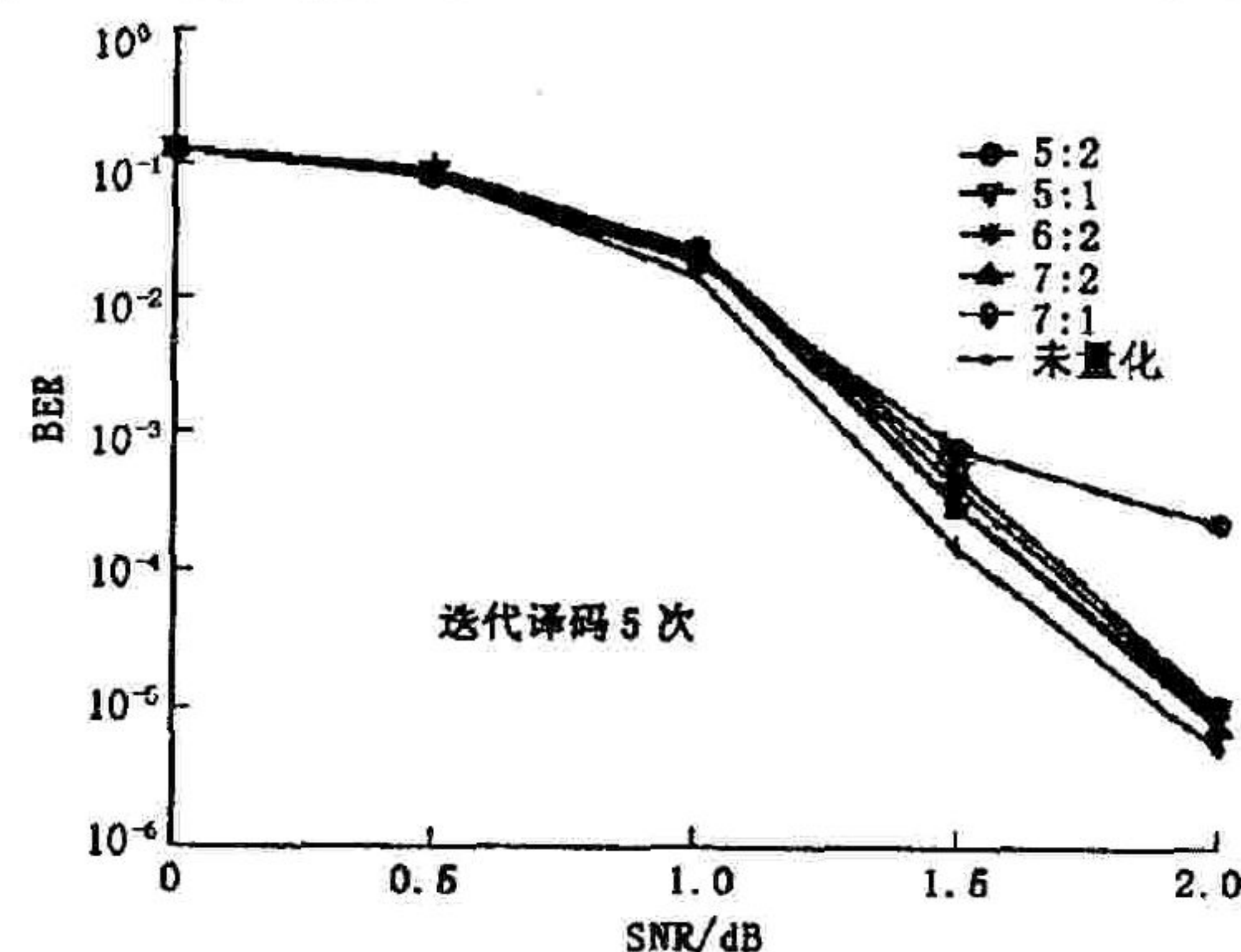


图 2 外信息量化对 Turbo 码性能的影响

用来表示小数部分即按 7:2 的定点量化方式就可以保证应有的译码性能。

3 状态度量的量化与重整

状态度量需要依据路径度量递推算出,而且随着递推的进行,状态度量值无限增大,如果不进行相应处理会导致数据溢出而造成译码错误。由于某一时刻所有的状态度量同时减去一个常数不影响正确译码,而且任两个状态度量之间的差值具有上下限^[7],为防止数据溢出,一种比较直观的解决方法是某一时刻将每一个状态度量都减去其中的最小值,但是这种方法在硬件实现时需要较多的比较器与减法器,并且会带来较大的译码延迟。一些文献^[8-9]沿用 Viterbi 译码算法关于避免路径度量溢出的取模归一化算法,该算法的缺陷是,在实现时需要对状态度量先进行位扩展,即对状态度量的表示字宽拓宽一个比特。

本文采用的是文献[10]给出的状态度量重整方法,见图3。 P_1 、 P_2 是两个正负门限值,对于我们选用的(13,15)编码器,完全可以将两个门限值 P_1 、 P_2 减少为一个正的 门限值 P_1 ,这是因为对于(13,15)RSC 码的栅格图而言,某一固定状态参加每一次状态度量递推运算的两个路径度量均互为相反数,又由于 Max 函数执行“取大”运算,故每一步递推状态度量都要累加一个非负值,唯一需要处理的是状态度量递推的初值。理论上,若编码器从零状态出发,在对数域内对应零状态的状态度量初值。在实现时通常分别赋为 0 和 $-(2^{n-1}-1)$ (n 为状态度量定点表示所需要的比特数)。最终产生用来判决的对数似然比来自对应译码比特的累加度量之差,因此我们将零状态的状态度量初值赋为 $2^{n-1}-1$,其它状态定为 0 不会影响译码性能,这样可以保证状态度量只能正向超出门限 P_1 ,而不会负向低于门限 P_2 ,从而只需设定一个门限值。在仿真时,我们对状态度量实行 8:2 量化,门限值设为 21.50。

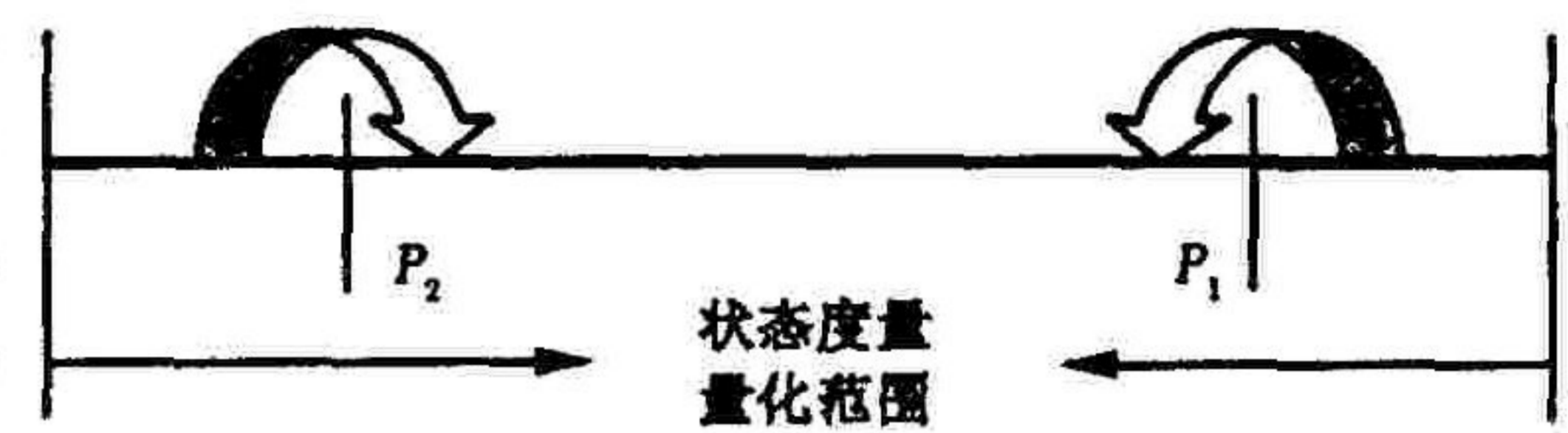


图3 状态度量重整示意图

4 查表运算的量化处理及量化性能分析

与 Max-Log-MAP 算法不同,Log-MAP 在实现时不仅是类似 Viterbi 中的加比选(ACS)结构,同时还有一个补偿值 $\ln(1+e^{-|x-y|})$ 的计算,即所谓的 ACSO 结构。对于补偿值通常采用查找表方法实现,路径度量与状态度量量化后查找表即基本确定,考察函数 $f(z) = \ln(1+e^{-|z|})$ 的值域,经过量化处理后如表2。

表2 查表运算的量化处理结果

$ x-y $	0.00	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	其它
$\ln(1+e^{- x-y })$	0.75	0.50	0.50	0.50	0.25	0.25	0.25	0.25	0.25	0.00

图4是所有数据按上述方案量化处理后的译码性能与未量化之前的性能比较,可以看出,按照本文的量化方式,量化损失增益不大于 0.15 dB。

5 结束语

Turbo 码以其低信噪比下优异的性能获得了广泛而持续的关注,在各类通信系统中都具有广阔的应用前景和巨大的应用潜力,目前对 Turbo 码的研究已进入应用阶段,对 Turbo 码编译码器的开发与实现是研究工作的热点之一。Turbo 译码为软输入软输出的迭代译码,因此其软数据的定点量化是 FPGA 开发与定点 DSP 实现的重要前提,本文基于 Log-MAP 译码算法,系统给出了 Turbo 迭代译码过程中所有数据的量化处理细节,选择的量化方案达到了性能与实现复杂度较好的折衷,可以用于高速率的 Turbo 译码器的开发与实现,为 Turbo 码工程应用提供了重要参考。

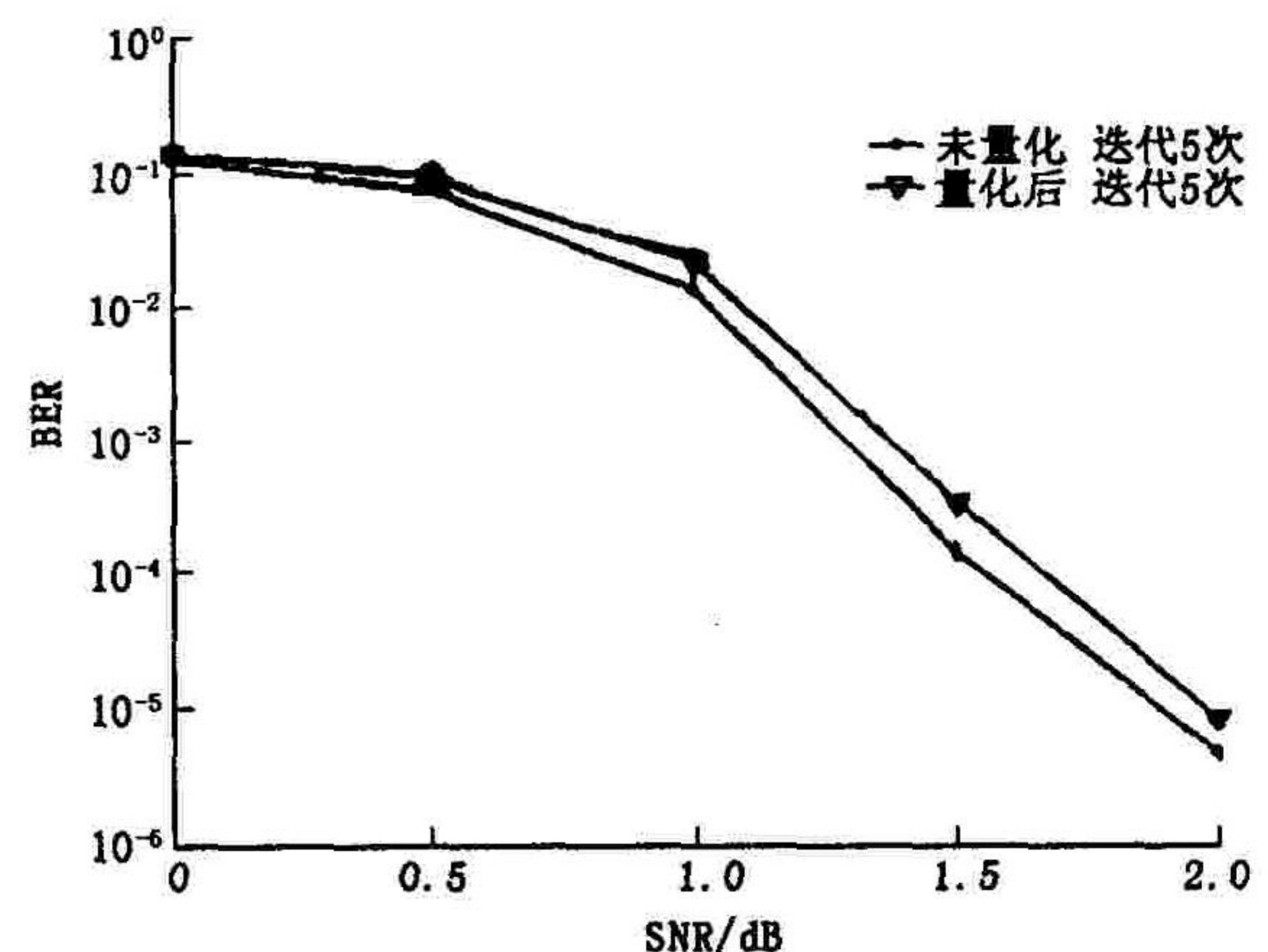


图4 量化前后的 Turbo 码性能比较