

用于 Turbo 迭代译码的 log-MAP 算法的简化¹

刘东华 唐朝京

(国防科技大学电子科学与工程学院 长沙 410073)

摘要 对用于 Turbo 迭代译码的对数最大后验概率 (log-MAP) 译码算法进行了分析和推导。根据 Turbo 编译码特点以及对格图中分支路径量度特性的分析, 得到了简化分支路径量度和外部信息计算的方法, 并就 log-MAP 算法中形如 $\ln(e^x + e^y)$ 的计算作了分析和化简, 进一步减小 log-MAP 算法的复杂性。最后给出仿真结果并进行了分析。

关键词 Turbo 码, log-MAP 算法, 迭代译码, 外部信息

中图分类号 TN911.21

1 前言

Turbo 码^[1]自 1993 年问世以来, 由于其接近 Shannon 极限的卓越性能而引起了世界范围内信息编码理论界的广泛兴趣。Turbo 码在译码过程中采用了迭代译码^[2]的思想, 其最优译码算法是修正的最大后验概率 (MAP) 算法^[3]。但 MAP 算法涉及到指数加法和实数递归乘法的运算, 复杂度非常之大。为实现采用最大后验概率算法的 Turbo 译码器, 人们提出了对数形式的 MAP 算法——log-MAP 算法^[4]。本文针对 log-MAP 算法用于 Turbo 译码的具体特点, 对译码格图中分支路径量度和外部信息的计算进行简化, 并从理论上解释了所做简化不会影响译码性能的原因。然后分析了 log-MAP 算法中 $\ln(e^x + e^y)$ 算式的计算并通过多项式拟合进行化简, 最后给出仿真结果和分析结论。

2 log-MAP 算法的推导

设编码器输入信息序列为 $d_1^N = \{d_1, d_2, \dots, d_N\}$, 输出码字为 $C_1^N = \{C_1, C_2, \dots, C_N\}$, 假定编码速率为 1/2, 记 k 时刻的输出码字为 $C_k = (c_k^1, c_k^2)$ 。码序列经过 BPSK 调制后得到发送信息序列 $Z_1^N = \{Z_1, Z_2, \dots, Z_N\}$, 其中 $Z_k = (z_k^1, z_k^2)$, 而且有 $z_k^i = 2c_k^i - 1, i = 1, 2$ 。经过信道传输后在接收端得到接收序列 $R_1^N = \{R_1, R_2, \dots, R_N\}$, 其中 $R_k = (x_k, y_k)$, 若参考信道为加性高斯白噪声 (AWGN) 信道, 则有 $x_k = z_k^1 + i_k, y_k = z_k^2 + q_k$, 其中 i_k 和 q_k 是均值为零, 方差为 σ^2 的独立同分布高斯随机变量。

2.1 MAP 算法

MAP 译码最终要得到的软判决输出 (即后验概率似然比) 为

$$\Lambda(d_k) = \log[P(d_k = 1|R_1^N)/P(d_k = 0|R_1^N)] \quad (1)$$

若 $\Lambda(d_k) > 0$, 则判决值为 $\hat{d}_k = 1$; 否则判决为 $\hat{d}_k = 0$ 。

为计算软输出对数概率似然比, 定义 k 时刻的几个联合概率和条件概率:

(1) 分支路径量度 $r(R_k, m', m)$

¹ 2000-06-05 收到, 2001-01-22 定稿
国家部级基金项目和校创新基金项目资助

$$\begin{aligned} r(R_k, m', m) &= P(S_k = m, R_k | S_{k-1} = m') = \sum_{Z_k} P(S_k = m, Z_k, R_k | S_{k-1} = m') \\ &= \sum_{Z_k} P(Z_k | S_k = m, S_{k-1} = m') P(S_k = m | S_{k-1} = m') P(R_k | Z_k) \end{aligned} \quad (2)$$

(2) 前向状态量度 $\alpha_k(m)$

$$\alpha_k(m) = P(S_k = m, R_1^k) = \sum_{m'} r(R_k, m', m) \alpha_{k-1}(m') \quad (3)$$

它是一个迭代计算过程, 初值设为 $\alpha_0(m=0) = 1, \alpha_0(m \neq 0) = 0$

(3) 后向状态量度 $\beta_k(m)$

$$\beta_k(m) = P(R_{k-1}^N | S_k = m) = \sum_{m'} r(R_{k+1}, m, m') \beta_{k+1}(m') \quad (4)$$

迭代初值为 $\beta_N(m=0) = 1, \beta_N(m \neq 0) = 0$

根据 $r(R_k, m', m), \alpha_k(m)$ 和 $\beta_k(m)$ 的定义, 可以得到 k 时刻的联合概率

$$\lambda_k(m', m) = P(S_{k-1} = m', S_k = m, R_1^N) = \alpha_{k-1}(m') r(R_k, m', m) \beta_k(m) \quad (5)$$

从而得到后验概率:

$$P(d_k | R_1^N) = \frac{1}{P(R_1^N)} \sum_{d_k=d:m} \sum_{m'} \lambda_k(m', m) \sum_{d_k=d:m} \sum_{m'} \alpha_{k-1}(m') r(R_k, m', m) \beta_k(m) \quad (6)$$

其中 $d = 0, 1$.

根据 (1) 式和 (6) 式, 可以得到最终似然比的表示形式为

$$\Lambda(d_k) = \log \frac{P(d_k = 1 | R_1^N)}{P(d_k = 0 | R_1^N)} = \log \frac{\sum_{d_k=1:m} \sum_{m'} \alpha_{k-1}(m') r(R_k, m', m) \beta_k(m)}{\sum_{d_k=0:m} \sum_{m'} \alpha_{k-1}(m') r(R_k, m', m) \beta_k(m)} \quad (7)$$

(1)-(7) 式组成了 MAP 算法.

2.2 log-MAP 算法

将 MAP 算法变换到对数域, 令 $L_{a,k}(d)$ 为 k 时刻送入译码器的先验信息似然比, $L_{s,k}(c^i)$ 为系统信息似然比, $L_{e,k}(d)$ 为外部信息似然比. 重新定义上述 3 个联合概率:

(1) 对数分支路径量度 $M(R_k, m', m) = \ln r(R_k, m', m)$

根据 (2) 式中 $r(R_k, m', m)$ 的表示, 当 k 时刻字为 Z_k 时, 若编码器由状态 m' 转移到 m , 则 $P(Z_k | S_k = m, S_{k-1} = m')$ 的值为 1, 否则为 0; 第 2 项为状态转移概率, 满足如下关系:

$$P(S_k = m | S_{k-1} = m') = \begin{cases} P(d_k = 1) = \frac{\exp(L_{a,k}(d))}{1 + \exp(L_{a,k}(d))}, & d_k = 1 \\ P(d_k = 0) = \frac{1}{1 + \exp(L_{a,k}(d))}, & d_k = 0 \end{cases} \quad (8)$$

AWGN 信道下 $r(R_k, m', m)$ 的第 3 项可写成

$$\begin{aligned}
 P(R_k|Z_k) &= P(x_k|d_k)P(y_k|d_k, S_{k-1} = m', S_k = m) \\
 &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x_k - z_k^1)^2}{2\sigma^2}\right] \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_k - z_k^2)^2}{2\sigma^2}\right] \quad (9)
 \end{aligned}$$

取对数, 有

$$\ln P(R_k|Z_k) = \frac{x_k z_k^1}{\sigma^2} + \frac{y_k z_k^2}{\sigma^2} = \frac{1}{\sigma^2} [x_k(2c_k^1 - 1) + y_k(2c_k^2 - 1)] \quad (10)$$

而

$$L_{s,k}(c^1) = \ln \frac{P(z_k^1 = 1|R_1^N)}{P(z_k^1 = -1|R_1^N)} = \ln \frac{P(x_k|z_k^1 = 1)}{P(x_k|z_k^1 = -1)} = \ln \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp(-(x_k - 1)^2/(2\sigma^2))}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp(-(x_k + 1)^2/(2\sigma^2))} = \frac{2}{\sigma^2} x_k \quad (11)$$

同理, 有

$$L_{s,k}(c^2) = (2/\sigma^2)y_k \quad (12)$$

将 (11), (12) 式代入 (10) 式, 得到分支路径量度的对数表示

$$\begin{aligned}
 M(R_k, m', m) &= \ln r(R_k, m', m) \\
 &= \begin{cases} L_{a,k}(d) - \ln(1 + \exp(L_{a,k}(d))) + \frac{1}{2}L_{s,k}(c^1) + \frac{1}{2}L_{s,k}(c^2)(2c_k^2 - 1), & d_k = 1 \\ -\ln(1 + \exp(L_{a,k}(d))) - \frac{1}{2}L_{s,k}(c^1) + \frac{1}{2}L_{s,k}(c^2)(2c_k^2 - 1), & d_k = 0 \end{cases} \quad (13)
 \end{aligned}$$

(2) 对数前向状态概率量度 $A_k(m) = \ln \alpha_k(m)$

$$A_k(m) = \ln \sum_{m'} r(R_m, m', m) \alpha_{k-1}(m') = \max_{m'}^* [A_{k-1}(m') + M(R_k, m', m)] \quad (14)$$

相应的初值为 $A_0(m=0) = 0$, $A_0(m \neq 0) = \infty$ 。其中 $\max^*[\]$ 定义为^[5]

$$\max^*(x_1, x_2, \dots, x_n) = \ln(e^{x_1} + e^{x_2} + \dots + e^{x_n}) \quad (15)$$

(3) 对数后向状态概率量度 $B_k(m) = \ln \beta_k(m)$

$$B_k(m) = \ln \sum_{m'} r(R_{k+1}, m, m') \beta_{k+1}(m') = \max_{m'}^* [B_{k+1}(m') + M(R_{k+1}, m, m')] \quad (16)$$

相应的初值为 $B_N(m=0) = 0$, $B_N(m \neq 0) = \infty$ 。 $\max^*[\]$ 的意义如上式。

由上述讨论, 根据 (6) 式, 可以写出信息比特后验概率的对数似然比:

$$\begin{aligned}
 \Lambda(d_k) &= \ln \sum_{d_k=1:m} \sum_{m'} d_{k-1}(m') r(R_k, m', m) \beta_k(m) - \ln \sum_{d_k=0:m} \sum_{m'} \alpha_{k-1}(m') r(R_k, m', m) \beta_k(m) \\
 &= \max_{d_k=1:m, m'}^* [A_{k-1}(m') + M(R_k, m', m) + B_k(m)] \\
 &\quad - \max_{d_k=0:m, m'}^* [A_{k-1}(m') + M(R_k, m', m) + B_k(m)] \quad (17)
 \end{aligned}$$

将 (13) 式代入 (17) 式, 提取公因子, 得

$$\begin{aligned} \Lambda(d_k) &= L_{a,k}(d) + L_{s,k}(c^1) + \max_{d_k=1:m,m'}^* \left[A_{k-1}(m') + \frac{1}{2} L_{s,k}(c^2)(2c_k^2 - 1) + B_k(m) \right] \\ &\quad - \max_{d_k=0:m,m'}^* \left[A_{k-1}(m') + \frac{1}{2} L_{s,k}(c^2)(2c_k^2 - 1) + B_k(m) \right] \end{aligned} \quad (18)$$

$\Lambda(d_k)$ 由三部分组成: 先验信息 $L_{a,k}(d)$ 、系统信息 $L_{s,k}(c^1)$ 和外部信息 (剩余部分)。外部信息 $L_{e,k}(d)$ 可表示为

$$\begin{aligned} L_{e,k}(d) &= \Lambda(d_k) - L_{a,k}(d) - L_{s,k}(c^1) \\ &= \max_{d_k=1:m,m'}^* \left[A_{k-1}(m') + \frac{1}{2} L_{s,k}(c^2)(2c_k^2 - 1) + B_k(m) \right] \\ &\quad - \max_{d_k=0:m,m'}^* \left[A_{k-1}(m') + \frac{1}{2} L_{s,k}(c^2)(2c_k^2 - 1) + B_k(m) \right] \end{aligned} \quad (19)$$

(15)-(19) 式组成了 log-MAP 算法。

3 对 log-MAP 算法的简化

考虑对上述 log-MAP 算法的简化。由分析可知, 算法的运算量主要集中在分支路径量度 $M(R_k, m', m)$ 和状态概率量度 $A_k(m)$ 和 $B_k(m)$ 的计算上, 而 $A_k(m)$ 和 $B_k(m)$ 的简化还涉及到对 (15) 式 $\max^*(\)$ 算式的简化。同时简化对 $\max^*(\)$ 的计算也可以在很大程度上简化 (18)、(19) 两式中对数似然比和外部信息的计算。下面主要考虑从这两方面简化 log-MAP 算法。

3.1 对 $M(R_k, m', m)$ 计算的简化

(1) 由 (13) 式可以看出, 无论信息比特的取值为 0 还是 1, 在 $M(R_k, m', m)$ 的表示式中, 总有一部分是不变量, 在计算中省略掉这部分常数对最终得到的似然比没有影响, 但这可以减少运算的复杂度。重新整理 (13) 式, 得到

$$\begin{aligned} M(R_k, m', m) &= \begin{cases} L_{a,k}(d) + c_k^1 L_{s,k}(c^1) + c_k^2 L_{s,k}(c^2) \\ \quad + [-\ln(1 + \exp(L_{a,k}(d))) - (1/2)L_{s,k}(c^1) - (1/2)L_{s,k}(c^2)], & d_k = 1 \\ c_k^1 L_{s,k}(c^1) + c_k^2 L_{s,k}(c^2) \\ \quad + [-\ln(1 + \exp(L_{a,k}(d))) - (1/2)L_{s,k}(c^1) - (1/2)L_{s,k}(c^2)], & d_k = 0 \end{cases} \\ &= d_k L_{a,k}(d) + c_k^1 L_{s,k}(c^1) + c_k^2 L_{s,k}(c^2) + C, \quad d_k \in \{0, 1\} \end{aligned} \quad (20)$$

其中常数部分:

$$C = -\ln(1 + \exp(L_{a,k}(d))) - L_{s,k}(c^1)/2 - L_{s,k}(c^2)/2 \quad (21)$$

与信息比特 d_k 的取值无关, 在计算时可省略。

(2) 考虑具体把 log-MAP 算法用于 Turbo 译码, 可以进一步简化计算。在传统的 Turbo 译码过程中, 要求将接收系统信息交织后也送入第二个成员译码器, 而实际上可以通过改变外部

信息的值,使之包含系统信息,这样就不必把系统信息送入第二个成员译码器。因此,对于编码速率为 $1/2$ 的成员码而言,可以把第二个成员码的编码速率看成是 $1/1$,即不传递系统信息。这时,对于两个成员译码器而言, $M(R_k, m', m)$ 的计算就有所区别:

对于第一个成员码,有

$$M_1(R_k, m', m) = d_k L_{1a,k}(d) + c_{1k}^1 L_{1s,k}(c^1) + c_{1k}^2 L_{1s,k}(c^2) \quad (22)$$

而对于第二个成员码,则有

$$M_2(R_k, m', m) = d_k L_{2a,k}(d) + c_{2k}^2 L_{2s,k}(c^2) \quad (23)$$

(3) 由于系统信息 $L_{s,k}(c^1)$ 仅包括在第一个成员译码器格图的分支路径量度 $M_1(R_k, m', m)$ 中,而不包括在 $M_2(R_k, m', m)$ 中,故需要修正外部信息的表示式。对于第一个成员译码器的软输出似然比不变:

$$\begin{aligned} \Lambda'_1(d_k) &= L'_{1a,k}(d) + L_{1s,k}(c^1) \\ &+ \max_{d_k=1:m,m'}^* \left[A_{1,k-1}(m') + \frac{1}{2} L_{1s,k}(c^2)(2c_{1k}^2 - 1) + B_{1,k}(m) \right] \\ &- \max_{d_k=0:m,m'}^* \left[A_{1,k-1}(m') + \frac{1}{2} L_{1s,k}(c^2)(2c_{1k}^2 - 1) + B_{1,k}(m) \right] \end{aligned} \quad (24)$$

由于在 $M_2(R_k, m', m)$ 中不包括 $L_{2s,k}(c^1)$,而 $L_{1s,k}(c^1)$ 应该是第一个成员译码器输出外部信息的一部分。因此,要把第二个成员译码器输入的系统信息作为先验信息的一部分。因此

$$\begin{aligned} L'_{1e,k}(d) &= \Lambda'_1(d_k) - L'_{1a,k}(d) \\ &= L_{1s,k}(c^1) + \max_{d_k=1:m,m'}^* \left[A_{1,k-1}(m') + \frac{1}{2} L_{1s,k}(c^2)(2c_{1k}^2 - 1) + B_{1,k}(m) \right] \\ &- \max_{d_k=0:m,m'}^* \left[A_{1,k-1}(m') + \frac{1}{2} L_{1s,k}(c^2)(2c_{1k}^2 - 1) + B_{1,k}(m) \right] \\ &= L_{1a,k}(d) + L_{1s,k}(c^1) \end{aligned} \quad (25)$$

这是第一个成员译码器输出的外部信息,交织后作为先验信息 $L_{2a,k}(d)$ 输入到第二个成员译码器。新的先验信息与原来的外部信息之间的关系为

$$L'_{2a,k}(d) = L'_{1a,\alpha(k)}(d) = L_{1a,\alpha(k)}(d) + L_{1s,\alpha(k)}(c^1) = L_{2a,k}(d) + L_{1s,\alpha(k)}(c^1) \quad (26)$$

它是原来的先验信息和系统信息之和。第二个成员译码器的软输出信息仅仅是在原来的表达式中去除掉 $L_{s,k}(c^1)$ 项,从而得到

$$\begin{aligned} \Lambda'_2(d_k) &= L'_{2a,k}(d) + \max_{d_k=1:m,m'}^* \left[A_{2,k-1}(m') + \frac{1}{2} L_{2s,k}(c^2)(2c_{2k}^2 - 1) + B_{2,k}(m) \right] \\ &- \max_{d_k=0:m,m'}^* \left[A_{2,k-1}(m') + \frac{1}{2} L_{2s,k}(c^2)(2c_{2k}^2 - 1) + B_{2,k}(m) \right] = \Lambda_2(d_k) \end{aligned} \quad (27)$$

第二个成员译码器输出的外部信息为

$$L'_{2e,k}(d) = \Lambda'_2(d_k) - L'_{2a,k}(d) = L_{2e,k}(d) \quad (28)$$

可以看到当把第二个成员编码器看作是码率为 1/1 的码时, 第二个成员译码器的外部信息不变. 亦即第一个成员译码器的先验信息不变, 即 $L'_{1a,k}(d) = L_{1a,k}(d)$. 同时两个成员译码器的软输出值 $\Lambda_1(d_k)$ 和 $\Lambda_2(d_k)$ 保持不变. 同时它们的外部信息可以写成统一的形式 $L'_{e,k}(d) = \Lambda(d_k) - L_{a,k}(d)$. 这样改进的结果是减少了运算量, 同时也简化了译码器的结构.

3.2 对 $\max^*()$ 计算的简化

根据 (15) 式中 $\max^*()$ 的定义, 变量个数为 2 时, 有

$$\max^*(x_1, x_2) = \max(x_1, x_2) + \ln(1 + e^{-|x_1 - x_2|}) = \max(x_1, x_2) + f_c(|x_1 - x_2|) \quad (29)$$

而三变量 $\max^*()$ 算式具有如下性质:

$$\max^*(x_1, x_2, x_3) = \max^*(x_1, \max^*(x_2, x_3)) = \max^*(\max^*(x_1, x_2), x_3) \quad (30)$$

因此, 多个变量的 $\max^*()$ 算式都可以化为两个变量的形式, 根据 (30) 式, 每计算一个被译比特的状态概率量度就需要 $2^v - 1$ 次 $\max^*()$ 运算, 而在整个 log-MAP 算法中计算一个被译比特的似然输出共需要 $4 \times (2^v - 1)$ 次 $\max^*()$ 运算, 其中 v 为编码寄存器数, 2^v 表示译码器所有可能状态. 由 (29) 式可见, $\max^*()$ 的计算量主要集中在对 $f_c(x)$ 的计算上, 因此考虑对 $f_c(x)$ 的简化. 现用多项式拟合的方法进行简化, 采用 5 次多项式拟合, 均方误差在 10^{-4} 量级. 拟合多项式为

$$f_c(x) \approx 6.5 \times 10^{-5} x^5 + 3.8 \times 10^{-4} x^4 - 1.8 \times 10^{-2} x^3 + 1.5 \times 10^{-2} x^2 - 5.1 \times 10^{-1} x + 6.9 \times 10^{-1} \quad (31)$$

通过仿真, 其计算时间比直接计算 $f_c(x)$ 要节省大约 20%.

另外, 由于 $f_c(x)$ 是递减函数, 随着变量值 x 的增加, 函数值减小, 当 $x > 7$ 时, $f_c(x)$ 的取值达到 10^{-4} 量级, 因此对于 $x > 7$ 的情况, 可以近似取 $f_c(x)$ 为 0. 即

$$f_c(x) \equiv 0, \quad x \geq 7 \quad (32)$$

而在 $0 < x < 7$ 时, 用 (31) 式计算 $f_c(x)$.

4 仿真结果及分析

在对计算分支路径量度 $M(R_k, m', m)$ 所做的简化中, 第一点只是将 $M(R_k, m', m)$ 表示式中与信息比特 d_k 取值无关的项省略; 第二点是省略掉重复输入的系统信息部分; 而最后一点是根据前两点简化后不同成员译码器的分支路径量度 $M(R_k, m', m)$ 表达式的变化来修正外部信息的表示式. 因此, 对分支路径量度 $M(R_k, m', m)$ 所做的三点简化不会使译码性能变差.

简化后的 log-MAP 算法与原来的 log-MAP 算法相比, 用于 Turbo 迭代译码的过程是一样的. 在简化的 log-MAP 算法中, 两个成员译码器的功能也保持不变. 但对于第二个成员译码器, 在原来的 log-MAP 算法中, 需要三个输入 ($L_{2a,k}(d)$, $L_{2s,k}(c^1)$ 和 $L_{2s,k}(c^2)$), 而作了简化以后, 仅需要两个输入 ($L_{2a,k}(d)$ 和 $L_{2s,k}(c^2)$). 从而使译码器的结构得到简化. 简化的 log-MAP 算法的软输出仅由两部分组成: 即新的先验信息和新的外部信息. 对于第一个成员译码器来说, 其先验信息不变, 但其外部信息则由原来的系统信息和原来的外部信息组成; 而对于第二个成员译码器, 其外部信息不变, 其先验信息是原来的系统信息和原来的先验信息之和. 因此, 作

了上述简化只是使译码输入输出的形式发生了改变, 而对于两个成员译码器的软输出没有任何影响, 因此, 对分支路径量度 $M(R_k, m', m)$ 计算的简化不会影响系统译码性能。

上述对状态概率量度计算的简化时, 由于引入了对 $f_c(x)$ 函数的近似计算, 所以对译码性能有一定的影响。计算机仿真结果如图 1 所示。其中 N 代表交织器的大小。由图 1 可以看出: 由于在对 $\max^*(\)$ 的计算过程中引入了近似计算, 所以简化的 log-MAP 算法性能要比标准的 log-MAP 算法性能稍差, 而且在不同的交织器长度、不同译码迭代次数和整个仿真信噪比 (SNR) 范围内都是如此。

在 SNR 较小时, log-MAP 算法的简化对性能的影响要比高 SNR 条件下小。这是因为低 SNR 下, 码字比特出现差错主要是由于信道噪声造成的, 与之相比, $\max^*(\)$ 近似计算所带来的错误影响则比较小。而在大 SNR 时信道噪声较小, 所引起的传输错误也较少, 因此, 近似计算的误差影响就成为额外错误比特的主要因素, 导致性能降低程度加大。

随着译码迭代次数的增加, $\max^*(\)$ 近似计算带来的性能降低也越多。这显然是因为在迭代过程中近似计算误差的累加所造成的。

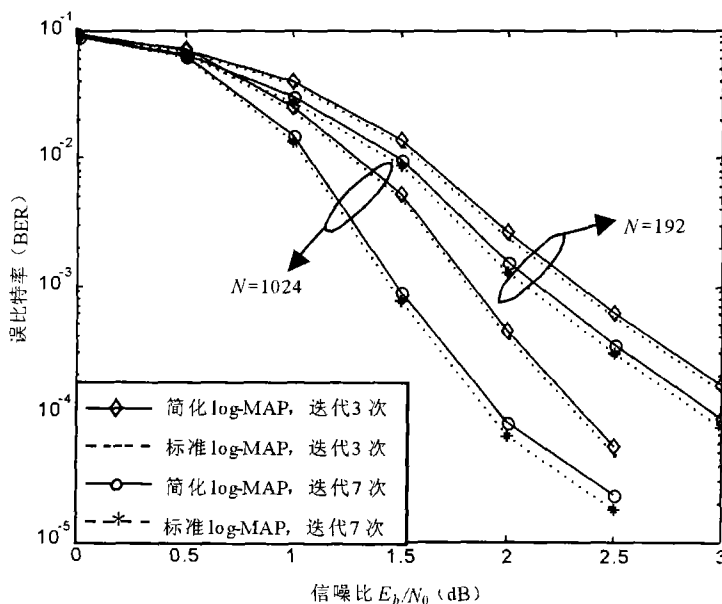


图 1 简化前后的 log-MAP 算法用于 Turbo 译码的性能比较曲线

5 结 论

Turbo 码因其卓越的译码性能而成为当今纠错编码研究的热点, 有关它在不同通信系统中的应用问题也在不断深入探讨。但由于译码的大运算量和延时大大限制了它的应用范围。本文针对 log-MAP 算法用于 Turbo 迭代译码这一特定条件, 从分支路径量度计算和状态概率量度计算两方面对算法进行了简化, 有效地减少了 log-MAP 算法的计算量和译码延时, 这对于工程上实现 log-MAP 译码器是非常有益的。

参 考 文 献

- [1] C. Berrou, A. Glavieux, P. Thitimajshima, Near Shannon limit error correcting coding and decoding: Turbo-codes(1), In ICC'93, Geneva, Switzerland, May, 1993, 1064-1070.
- [2] S. Benedetto, D. Divsalar, G. Montorsi, F. Pollara, Soft-output decoding algorithms in iterative decoding of turbo codes, TDA Progress Report 42-119, February, 1996, 63-87.
- [3] L. R. Bahl, J. Cocke, F. Jeinek, J. Raviv, Optimal decoding of linear codes for minimizing symbol error rate, IEEE Trans. on IT, 1974, IT-20(3), 248-287.
- [4] S. Benedetto, D. Divsalar, G. Montorsi, F. Pollara, A soft-input soft-output maximum a posteriori (MAP) module to decode parallel and serial concatenated codes, TDA Progress Report 42-127, November 1996, 1-20.
- [5] P. Robertson, E. Vilebrum, P. Hoeher, A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain, In ICC'95, Seattle, Washington, June 1995, 1009-1013.

SIMPLIFICATION OF LOG-MAP ALGORITHM USED FOR
TURBO ITERATIVE DECODING

Liu Donghua Tang Chaojing

(School of Electron. Sci. and Eng., Nat. Univ. of Defence Tech, Changsha 410073, China)

Abstract In this paper, logarithm MAP algorithm used for turbo iterative decoding is analyzed and deduced. Based on the characteristic of turbo coding and decoding and the analysis of branch metric in the trellis, methods of simplifying computation of branch metric and extrinsic information are obtained. And then it is done with the computation of $\ln(e^x + e^y)$ in the log-MAP algorithm to further simplify the complexity of log-MAP. Finally, simulation results and analysis are carried out.

Key words Turbo code, Log-MAP algorithm, Iterative decoding, Extrinsic information

刘东华: 男, 1976 年生, 博士生, 研究方向为纠错编码理论及移动通信系统仿真等.

唐朝京: 男, 1962 年生, 国防科技大学电子科学与工程学院副院长, 教授, 主要研究方向为信息论与编码、保密通信及多媒体会议电视系统等.