

# 基于改进型 SOVA 的 Turbo 译码

崔 龙, 凌 聪, 吴晓富  
(南京通信工程学院, 江苏南京 210016)

**摘 要:** 第三代移动通信系统 IMT-2000 的高速率业务倾向于选择 Turbo 码, 这就要求采用低时延、低复杂度的迭代译码技术, 主要是软输出 Viterbi 算法(SOVA)和 Max Log MAP 算法. 在先验等概和无限译码深度条件下, 已证明略加修改的 SOVA 等效于 Max Log MAP 算法. 由于在迭代译码中, 先验概率须不断更新, 本文证明了在存在先验概率的条件下改进型 SOVA 与 Max Log MAP 也是等效的, 并讨论了有限译码深度限制下改进型 SOVA 与滑动窗口 Max Log MAP 算法的等效性.

**关键词:** 软输出 Viterbi 算法(SOVA); 最大后验概率(MAP)译码; Max Log MAP 译码

**中图分类号:** TN911.2 **文献标识码:** A **文章编号:** 0372-2112(2001)07-0923-04

## Modified SOVA-Based Turbo Decoding

CUI Long, LING Cong, WU Xiaofu

(Nanjing Institute of Communications Engineering, Nanjing, Jiangsu 210016, China)

**Abstract:** Turbo codes are preferred for high rate data services in the third generation mobile communication system IMT-2000. Thus low decoding delay and low complexity iterative decoding are of paramount importance. The modified soft output Viterbi algorithm(SOVA) and the Max Log MAP algorithm are such decoding techniques. The equivalence of the modified SOVA and Max Log MAP algorithm has been proven under the condition of no a priori information and infinite decoding depth. Since the a priori information has to be updated in iterative decoding, we prove their equivalence when there exists a priori information in the paper. We also discuss the relationship between the sliding window Max Log MAP algorithm and the modified SOVA for finite decoding depth.

**Key words:** SOVA; MAP decoding; Max Log MAP decoding

### 1 引言

Viterbi 算法(VA) 能实现白噪声下的最大似然译码, 而且其实现比较简单, 因此被广泛应用于数字通信. 和 Viterbi 算法相比, 最大后验概率(MAP)译码<sup>[1]</sup>的优点在于能以软值的形式输出每个符号的后验概率, 这对一个要进行后续处理的通信系统的全局性能有重要意义. 不幸的是, MAP 算法的复杂度比 VA 要高得多, 不仅需要在格状图上进行前向和后向的递归, 而且须要知道噪声方差. 为了减少复杂度, 出现了两种次最佳算法: Max Log MAP 算法和软输出 Viterbi 算法(SOVA), 它们都不须要知道噪声方差. Max Log MAP 算法是 MAP 算法的一种近似, 也需要在格状图上进行前向和后向的递归, 但度量的计算被简化; SOVA 有两种不同的更新规则, 分别由 Hagenauer<sup>[2]</sup>和 Fosserier<sup>[3]</sup>提出, 本文用原始 SOVA 和改进型 SOVA 来区分它们. 原始 SOVA 省略了一些更新步骤, 因此易于实现, 但性能较差. 改进型 SOVA 是原始 SOVA 的一种改进形式, 在文[3]中, 已经证明了在发送符号先验等概和无限译码深度的条件下, 改进型 SOVA 与 Max Log MAP 算法是等效的. 文献[4]将这种等效关系推广到多元码, 并给出了更简洁的证明.

越来越多的第三代移动通信系统 IMT-2000 建议在高速率业务中采用了 Turbo 码. 高速率业务的每帧数据最多可能包含几十 K 信息比特, 比如 CDMA2000<sup>[9]</sup> 最长的帧有约 20K 比特, 采用了 8 状态子码, 假设量化为 8 比特, 则仅存储前向递归度量就需多于 1M 比特的 RAM, 此时直接采用 MAP 算法无论在计算复杂度上还是在存储要求上都存在很大的问题. 因此研究低时延、低复杂度的迭代译码很有实用价值, 如 SOVA 和 Max Log MAP 算法, 为了降低译码时延, 人们提出了滑动窗口 Max Log MAP 算法<sup>[5]</sup>. 本文的目的是以低时延、低复杂度的迭代译码为背景, 在文献[3, 4]的基础上分析这些算法之间的关系.

第二节分析存在先验信息时, Max Log MAP 算法软输出的几何意义; 第三节证明在已知先验概率和无限译码深度的条件下, 改进型 SOVA 与 Max Log MAP 算法也是等效的, 而且有限译码深度限制下改进型 SOVA 与 Max Log MAP 近似等效于滑动窗口 Max Log MAP 算法; 第四节通过计算机模拟, 给出改进型 SOVA、Max Log MAP 算法和滑动窗口 Max Log MAP 算法的迭代译码性能, 并比较它们的优缺点.

## 2 存在先验信息时 Max-Log MAP 算法软输出的几 何意义

假设信息序列  $u = (u_1, u_2, \dots, u_k)$  经过编码, 映射为发送码字  $c = s(u) = (c_1, c_2, \dots, c_N)$ , 其中  $c_i \in \{-1, +1\}$ ,  $i = 1, 2, \dots, N$ ;  $u_j \in \{0, 1\}$ ,  $j = 1, 2, \dots, K$ . 经过加性高斯白噪声 (AWGN) 信道, 接收矢量为  $r = c + n$ , 式中  $n$  是一个  $N$  维的高斯随机矢量, 其均值为 0, 协方差矩阵为  $N_0/2I_N$ . MAP 算法输出每个信息比特的对数似然比 (LLR)

$$\Lambda_j = \ln \left[ \frac{\sum_{c: s(u); u_j=1} P(c|r)}{\sum_{c: s(u); u_j=0} P(c|r)} \right] \quad (1)$$

而 Max-Log MAP 算法在 MAP 算法的基础上应用近似关系

$$\ln \left( \sum_{i=1}^N \delta_i \right) = \ln \left( \max_{i \in \{1, \dots, N\}} \{\delta_i\} \right), \text{ 简化输出的 LLR 为}$$

$$\Lambda_j \approx \ln \left[ \max_{c: s(u); u_j=1} P(c|r) \right] - \ln \left[ \max_{c: s(u); u_j=0} P(c|r) \right] \quad (2)$$

据文献[6]有,  $P(u_j = \pm 1) = A_j \cdot e^{L(u_j) \cdot u_j/2}$  ( $A_j$  是与  $u_j$  无关的常数), 则得到  $P(c) = A \cdot e^{L(c) \cdot c/2}$ , 其中  $A = \prod_{j=1}^K A_j$  是一个与  $c$  无关的常数; “ $\cdot$ ” 表示矢量的内积;  $L(c)$  为一个  $N$  维矢量, 把它的分量定义为

$$L(c_i) = \begin{cases} L(u_i), & c_i \text{ 为信息位} \\ 0, & c_i \text{ 为校验位} \end{cases} \quad i = 1, 2, \dots, N.$$

又因为  $P(r)$  与  $c$  无关, 则有

$$P(c|r) = P(r|c) \cdot P(c) / P(r) \\ \propto \exp[-\|r-c\|^2 / N_0] \cdot \exp[L(c) \cdot c/2]$$

根据上式, 式(2)中右边第一项等效于

$$\ln \left[ \max_{c: s(u); u_j=1} P(c|r) \right] \Leftrightarrow \max_{c: s(u); u_j=1} [-\|r-c\|^2 / N_0 + L(c) \cdot c/2] \\ \Leftrightarrow \min_{c: s(u); u_j=1} [L_c \cdot \|r-c\|^2 - 2L(c) \cdot c] \quad (3)$$

式中  $L_c = 4/N_0$  是一个与  $c$  无关的正常数. 附录中证明了, 当发送端的信息比特先验等概时, 令  $L_c = 1$  并不影响 Max-Log MAP 译码, 只是它的软值相差一个常系数. 又因为  $\|c\|^2$ 、 $\|L(c)\|^2$ 、 $r \cdot L(c)$  与具体的码字  $c$  无关\*, 所以上式可等效为

$$\ln \left[ \max_{c: s(u); u_j=1} P(c|r) \right] \Leftrightarrow \min_{c: s(u); u_j=1} [\|r-c\|^2 - 2L(c) \cdot c] \\ \Leftrightarrow \min_{c: s(u); u_j=1} \|r+L(c)-c\|^2$$

代入式(2)可得

$$\Lambda_j \propto \min_{c: s(u); u_j=1} \|r+L(c)-c\|^2 - \min_{c: s(u); u_j=0} \|r+L(c)-c\|^2 \quad (4)$$

由式(4)可以看出, 若把  $r+L(c)$  看作译码器的输入, 式(4)中右边平方欧氏距离较小的那项将对应于最大似然码字, 而另一项对应于在第  $j$  位与最大似然码字不同的最相似码字. 因此, Max-Log MAP 的硬判码字是对应输入为  $r+L(c)$  的最大似然码字, 而输出 LLR 与上述平方欧氏距离的差成比例. 这就是迭代译码中 Max-Log MAP 的几何解释. 很明显, 当不存在先验信息, 即  $L(c) = 0$  时, 式(4)变为

$$\Lambda_j \propto \min_{c: s(u); u_j=1} \|r-c\|^2 - \min_{c: s(u); u_j=0} \|r-c\|^2$$

与文献[3]相符.

## 3 存在先验信息时改进型 SOVA 与 Max-Log MAP 的等效性

为了便于讨论, 本文考虑码率为  $1/N$  (或由它删截而成的码率为  $K/N$ ) 的二进制递归系统卷积 (RSC) 码, 二者的格状图相同, 每个状态有两条输入支路, 分别对应不同的信息比特  $u_j \in \{1, 0\}$ . 编码器的记忆长度是  $v$ . 对其它码率的情况可以类推.

第  $k$  个译码时刻, 和 Viterbi 算法一样, 改进型 SOVA 存储状态  $s_k$  当前的路径度量  $\Gamma(s_k)$  和  $\delta+1$  个最新的信息位  $\hat{u}(S_k) = \{\hat{u}_{k-\delta}(S_k)\}$ , 这里  $\delta$  为改进型 SOVA 的译码深度. 此外, 改进型 SOVA 还存储对应于  $\hat{u}(S_k)$  的可靠性度量  $L(S_k) = \{L_{k-\delta}(S_k), \dots, L(S_k)\}$ .

在第  $k+1$  个译码时刻, 改进型 SOVA 分两步实现. 第一步和 Viterbi 算法一样, 计算进入  $S_{k+1}$  的两条支路的路径度量  $\Gamma(S_k^1, S_{k+1})$ ,  $\Gamma(S_k^2, S_{k+1})$ , 这里

$$\Gamma(S_k^i, S_{k+1}) = \Gamma(S_k^i) + \sum_{n=1}^N c_{k+1, n}^i \cdot r_{k+1, n} + c_{k+1, 1}^i \cdot L(c_{k+1}), \quad i = 1, 2 \quad (5)$$

取  $\Gamma(S_{k+1}) = \min_{i \in \{1, 2\}} \{\Gamma(S_k^i, S_{k+1})\}$ , 确定硬的判决结果  $\hat{u}_{k+1}(S_{k+1})$ , 更新  $\hat{u}(S_{k+1})$ .

第二步是更新可靠性度量  $L(S_{k+1})$ . 令  $L_{k+1}(S_{k+1}) = \Delta = |\Gamma(S_k^1, S_{k+1}) - \Gamma(S_k^2, S_{k+1})|$ , 定义  $L(S_k^1) = \{L_{k-\delta}^1, \dots, L_k^1\}$ ,  $L(S_k^2) = \{L_{k-\delta}^2, \dots, L_k^2\}$  为进入状态  $S_{k+1}$  的两条支路的始发状态  $S_k^1, S_k^2$  的可靠性度量, 不妨设发自  $S_k^1$  的支路为存活路径. 对应地,  $\hat{u}(s_k^1) = \{\hat{u}_{k-\delta}^1, \dots, \hat{u}_k^1\}$ ,  $\hat{u}(s_k^2) = \{\hat{u}_{k-\delta}^2, \dots, \hat{u}_k^2\}$  为这两条路径的最新  $\delta+1$  个判决值.  $L(S_{k+1})$  的更新规则为<sup>[3]</sup>

$$L_j(S_{k+1}) = \begin{cases} \min\{\Delta, L_j^1\}, & \hat{u}_j^2 \neq \hat{u}_j^1 \\ \min\{\Delta + L_j^2, L_j^1\}, & \hat{u}_j^2 = \hat{u}_j^1 \end{cases}, \quad j = k-\delta+1, \dots, k \quad (6)$$

而原始 SOVA 只考虑了上式的第一种情况, 对于第二种情况则保持  $L_j^1$  不变.

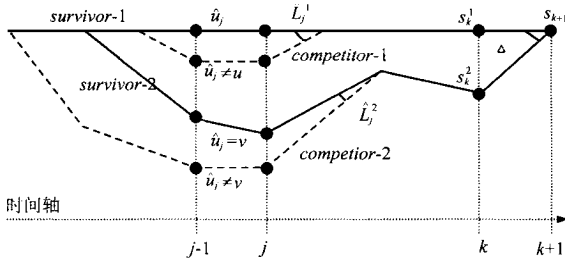
结论 1 存在先验信息时, 改进型 SOVA 与 Max-Log MAP 算法是等效的.

证明 要证明改进型 SOVA 与 Max-Log MAP 算法是等效的, 就是要证明对于这两个算法, 相同的输入导致相同的输出; 即要证明改进型 SOVA 的硬判码字也是对应输入为  $r+L(c)$  的最大似然码字, 而输出的第  $j$  个软值正比于最大似然码字和第  $j$  位与最大似然码字不同的最相似码字的平方欧氏距离之差.

首先, 参照式(5), Viterbi 算法使得度量  $r \cdot c + L(c) \cdot c$  最大, 即, 使得  $\|r+L(c)-c\|^2$  最小, 所以在译码最后时刻, 改进型 SOVA 的硬判码字是对应输入为  $r+L(c)$  的最大似然

\*  $L(u_j)$  是上一级译码的软输出, 表示本级译码第  $i$  位的先验信息, 与  $u_i = 1$  还是  $u_i = 0$  无关, 故  $L(c)$  与  $c$  无关.

码字.

图 1 改进型 SOVA 的格状图(图中  $u, v \in (0, 1)$ )

其次,由图 1 所示的格状图,用数学归纳法.假设在第  $k$  个译码时刻,  $L_j^1, L_j^2$  分别表示状态  $S_k^1, S_k^2$  的存活路径(Survivor 1, Survivor 2)与最相似路径(competitor 1, competitor 2)的路径度量差的绝对值,而且这两条路径在第  $j$  位的判决值不同.则在第  $k+1$  个译码时刻,状态  $S_{k+1}$  的存活路径必为 Survivor-1 和 Survivor-2 之一,不妨设为 Survivor-1, 其在第  $j$  位的判决值为  $u$ . 则到达状态  $S_{k+1}$  且满足  $\hat{u}_j \neq u$  的最相似路径是 survivor 2, competitor 1, competitor 2 三者之一. 否则,如果存在另外一条到达状态  $S_{k+1}$  且比这三条路径的度量更大的路径,由图 1 可知,此路径必须经过  $S_k^1$  或  $S_k^2$ , 则它比延伸前的 survivor 2 或 competitor 1, competitor 2 有更大的路径度量,这与 survivor 2, competitor 1, competitor 2 的定义矛盾.在图 1 中,如果  $u \neq v$  则最相似路径在 competitor 1, survivor 2 中选取,此时存活路径与第  $j$  位不同的最相似路径的度量差的绝对值为  $\min\{\Delta, L_j^1\}$ ; 如果  $u = v$ , 则最相似路径在 competitor 1, competitor 2 中选取,此时存活路径与第  $j$  位不同的最相似路径的度量差的绝对值为  $\min\{\Delta + L_j^2, L_j^1\}$ . 这正是改进型 SOVA 的更新规则,  $L_j(S_{k+1})$  也表示了状态  $S_{k+1}$  的存活路径与第  $j$  位不同的最相似路径的平方欧氏距离之差.所以改进型 SOVA 式(5)的度量计算和式(6)的更新规则必然产生 Max Log MAP 算法式(4)的软输出. □

应该指出的是,上述改进型 SOVA 与 Max Log MAP 算法的等效是有条件的,其条件是改进型 SOVA 的译码深度等于码长,即无限译码深度,这样将导致很大的译码时延、存储和复杂度.

为了降低译码时延,实际情况中 SOVA 总是与 Viterbi 算法一样,以一定的译码深度输出软值,这时它的译码时延就是译码深度  $\delta$ . 文[5]中提出了滑动窗口 Max Log MAP (SW-Max Log MAP) 算法,它的前向递归与 Max Log MAP 算法完全一样,但逆向递归不是从最后时刻才开始.对于窗口宽度为  $\delta$  的 SW-Max Log MAP 来说,第  $k$  个 LLR 由从时刻 0 到时刻  $k$  为止前向递归和从时刻  $k+\delta$  到时刻  $k$  为止的逆向递归产生,其译码时延也是  $\delta$ ,如图 2 所示.那么改进型 SOVA 与 SW-Max Log MAP 的关系是怎样的呢?由于两种算法的递归在时间上是不同的,文献[4]误认为它们的功能是不同的.下面将证明,这两种算法是近似等效的.

结论 2 译码深度为  $\delta$  的改进型 SOVA 与窗口宽度为  $\delta$  的 Max Log MAP 是近似等效的.

证明 参见图 2. 图中画出了四种算法的递归在时间上

的对比.

首先比较译码深度为  $\delta$  的改进型 SOVA 和无限译码深度(在图 2 中相当于译码深度为  $k+\delta$ )的改进型 SOVA. 注意到改进型 SOVA 的第  $k$  个 LLR 的迭代(6)只与时刻  $k$  之后的  $\Delta$  有关,因此,译码深度截短为  $\delta$  只是使得时刻  $k$  之前的软值与无限译码深度的改进型 SOVA 不同(事实上无限深度的改进型 SOVA 的这些软值更为精确),但就第  $k$  个(和  $k$  之后的) LLR 来说,二者输出的软值是相同的.

再考虑 SW-Max Log MAP,很明显,就第  $k$  个的 LLR 而言,它与 SW-Max Log MAP 的递归完全相同,故而二者第  $k$  个 LLR 相同.二者的不同点是时刻  $k$  之前的软值不同(事实上 Max Log MAP 算法的这些软值更为精确).

文献[3, 4]已证明,无限译码深度改进型 SOVA 与长度为  $k+\delta$  的 SW-Max Log MAP 算法等效.所以,就第  $k$  个 LLR 而言,译码深度为  $\delta$  的改进型 SOVA 与窗口宽度为  $\delta$  的 SW-Max Log MAP 是近似等效的. □

须要说明的一点是,对于 Max Log MAP 译码和窗口宽度等于码长的改进型 SOVA 译码,可以在  $k$  个信息比特的末尾添加  $v$  个结尾比特,使编码器的最终状态归零,从而使 SOVA 的软输出唯一,二者是严格等效的.但是在时延为  $\delta$  的算法中,SW-Max Log MAP 的逆向递归初始状态无法确定,只能假设为均匀分布;而 SOVA 的软输出有  $2^v$  种选择,我们选择度量  $\Gamma(S_{k+1})$  最大的状态所对应的软值作为输出 LLR.前者考虑了所有的状态,后者只考虑了最大似然状态,所以改进型 SOVA 的性能有微小的差异. □

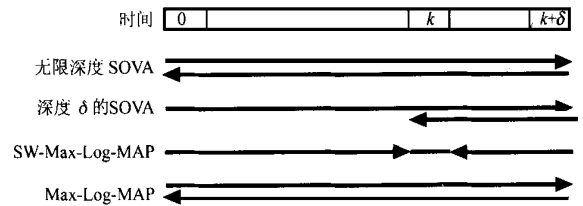


图 2 四种算法的递归在时间上的对比

#### 4 仿真结果及比较

图 3 模拟了译码深度  $\delta$  取 32 时,分别采用原始 SOVA、改进型 SOVA 和 SW-Max Log MAP 算法进行译码的 Turbo 码性能.作为比较,图中还给出了 Max Log MAP 算法的性能曲线.模拟时 Turbo 码的子码是 16 状态的(31, 33)码,帧长  $N = 1024$ ,没有任何结尾比特,伪随机交织,对两个子码编码器的校验位进行删截,使得总的码率  $R = 1/2$ ,采用 8 次迭代译码.可以看出,由于原始 SOVA 省略了一些更新步骤,其性能比改进型 SOVA 大约差 0.5 dB,而改进型 SOVA 的性能与 SW-Max Log MAP 算法基本相同,验证了它们之间的近似等效性.由于存在有限译码深度的限制,它们的软输出无法做到完全等同,从而它们的性能略有差异.图中的曲线显示改进型 SOVA 的性能略逊于 SW-Max Log MAP,性能差大约为 0.1 dB.模拟时还发现,对于改进型 SOVA,  $\delta$  的取值越大,其性能就越接近于 Max Log MAP,但改善越来越少,  $\delta = 32$  是性能与复杂度较好的折衷.

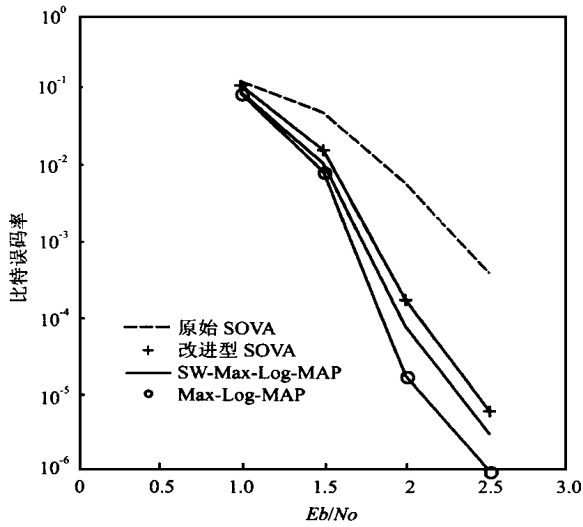


图3  $\delta = 32$  时各算法的 Turbo 译码性能曲线

各种算法每输出一个 LRR 的运算复杂度比较如表 1。假设取大操作、加法、位比较三种操作是等复杂度的, 则当  $v = 4$ ,  $\delta = 8 \times v = 32$  时, 原始 SOVA 的复杂度比 Max-Log-MAP 算法复杂度低一半, 而改进型 SOVA 的复杂度比原始 SOVA 高, 与 SW-Max-Log-MAP 算法基本相同。

存储复杂度方面, 改进型 SOVA 需要一个  $\delta \times 2^v$  的存储矩阵用于存储 LLR, 而 SW-Max-Log-MAP 仅需  $2^v$  个存储单元用于存储逆向递归度量。但由式(6)可知, 改进型 SOVA 的  $\delta$  级度量更新中, 各级之间是相互独立的, 可以并行实现, 而 SW-Max-Log-MAP 的  $\delta$  级逆向度量计算则得逐级串行实现。

表 1 运算复杂度对照表

	取大操作	加法	位比较
Max-Log-MAP <sup>[7]</sup>	$5 \times 2^v - 2$	$10 \times 2^v + 11$	0
SW-Max-Log-MAP	$(\delta + 5) \times 2^v$	$(3 \times \delta + 7) \times 2^v$	0
原始 SOVA <sup>[7]</sup>	$\delta/2 + 2^v$	$2 \times 2^v + 8$	$\delta$
改进型 SOVA	$(\delta + 1) \times 2^v$	$\delta \times 2^v$	$\delta \times 2^v$

## 5 结论

本文证明了在已知先验概率的条件下, 改进型 SOVA 与 Max-Log-MAP 算法的近似等效性, 论证了有限译码深度限制下改进型 SOVA 与滑动窗口 Max-Log-MAP 算法的近似等效性。迭代译码中, 大量中间计算结果的存储在硬件实现时是一个关键的问题, 改进型 SOVA 比较适合帧长很长的场合。由于现代芯片的集成度越来越高, 硬件实现对复杂度的要求有所降低, 所以在用硬件实现 Turbo 码的译码器时, 对算法的性能、计算复杂度和时延要综合考虑。如果系统要求有很低的时延, 而且芯片(如 FPGA)并行计算能力较强, 则改进型 SOVA 似乎是一个比较好的选择。

### 附录

文献[8]证明了对于 SOVA 译码, 令  $L_c = 1$  不影响译码, 这里的证明是类似的, Max-Log-MAP 算法第  $i$  级译码的输出  $L^i(\hat{u}_k)$  可分成两部分: 一部分是内信息  $L_i^i(\hat{u}_k)$ , 由信道输入  $L_c \cdot r$  和上级译码产生的先验信息  $L^i(u_k)$  组成; 另一部分是外

信息  $L_E^i(\hat{u}_k)$ , 它作为下一级译码的先验信息, 所以在下一级译码前要计算  $L_E^i(\hat{u}_k) = L^i(\hat{u}_k) - L^i(u_k)$ 。观察一下式(3), 由于在第一级译码的先验信息  $L^1(u_k) = 0$ ,  $L_c$  可以从  $\min\{\cdot\}$  中提出来, 最后, 它成为输出  $L^1(\hat{u}_k)$  的一个正的常数因子, 则第二级译码的先验信息  $L^2(u_k) = L^1(\hat{u}_k)$ , 也有因子  $L_c$ 。这样, 在第二级,  $L^2(u_k)$ 、 $L_c \cdot r$  都具有因子  $L_c$ , 在以后各级也具有因子  $L_c$ 。所以, 为简化计算, 令  $L_c = 1$  不会影响译码性能。

### 参考文献:

- [1] C Berrou, A Glavieux. Near optimum error correction coding and decoding: Turbo codes [J]. IEEE Trans. Commun., Oct. 1996, 44: 1261-1271.
- [2] J Hagenauer, P Hoher. A Viterbi algorithm with Soft decision outputs and its applications [A]. in Proc. Globecom, Dallas, TX [C], 1989: 1680-1686.
- [3] M P C Fossorier, F Burkert, S Lin, J Hagenauer. On the equivalence between SOVA and Max-Log-MAP decoding [J]. IEEE Commun. Lett., May 1998, 2: 137-139.
- [4] Ling Cong, Wu Xiaofu, and Yi Xiaoxin. On SOVA for nonbinary codes [J]. IEEE Commun. Lett., Dec. 1999, 3: 335-337.
- [5] S Benedetto, D Divsalar, G Montorsi, F Pollara. Algorithm for continuous decoding of turbo codes [J]. Electron. Lett., Feb. 1996, 32: 314-315.
- [6] J Hagenauer, E Offer, L Papke. Iterative decoding of binary block and convolutional codes [J]. IEEE Trans. Inform. Theory, Mar, 1996, 42: 429-445.
- [7] P Robertson, P Hoher, E Villebrun. Optimal and sub-optimal maximum a posteriori algorithm suitable for turbo decoding [J]. Europ. Trans. Telecom., Mar.-Apr. 1997, 8: 119-125.
- [8] L Ling, R S Cheng. Improvements in SOVA-based decoding for turbo codes [A]. in Proc. ICC '97 [C], 1473-1478.
- [9] TTA/EIA/IS 2000. 2 (Ballot Version) [S]. Physical layer Standard for cdma2000 spread spectrum systems. 2-36.

### 作者简介:



崔龙男, 1976年出生于湖北, 1998年毕业于南京通信工程学院无线通信系, 获学士学位, 2001年获该院通信与信息系统专业硕士学位, 研究方向是编码和迭代处理技术, 主要从事 Turbo 译码器的研发工作。



凌聪男, 1974年出生于安徽, 1995年毕业于南京通信工程学院无线通信系, 获学士学位, 1997年获该院通信与信息系统专业硕士学位, 现为该院讲师, 研究兴趣为无线通信, 重点是扩频通信、编码和迭代处理技术、时空调制等, 在国内外核心期刊发表论文十多篇, 主持第四代移动通信预研等多项基金项目。