

## 一种新的基-4SOVA 译码算法

任德锋\* 葛建华 王勇 宋英杰

(西安电子科技大学综合业务网理论与关键技术国家重点实验室 西安 710071)

**摘要:** SOVA (Soft Output Viterbi Algorithm)类算法因其译码时延远低于 MAP 类算法已成为 Turbo 码的实用译码算法, 为了进一步减小译码延迟, 提高译码速度, 该文在简单分析基-4Max-Log-MAP 算法的基础上, 提出了一种新的基-4SOVA 算法, 并进行了完整的数学推导。该算法的关键是提出了一种新的可信度更新方法, 可实现编码网格图中两步状态转移合并后的可信度的更新。仿真结果表明, 新算法的误码率性能非常接近基-4Max-Log-MAP 算法, 当自适应引入外信息系数后, 逼近基-4MAP。而且新算法具有译码延迟小、存储资源占用少等优点, 达到了与计算复杂度的良好折中。

**关键词:** 迭代译码; Turbo 码; 卷积 Turbo 码; 基-4 软输出维特比算法; 最大后验概率算法

**中图分类号:** TN911.22

**文献标识码:** A

**文章编号:** 1009-5896(2011)08-1963-06

**DOI:** 10.3724/SP.J.1146.2010.01379

## A Novel Radix-4SOVA Algorithm

Ren De-feng Ge Jian-hua Wang Yong Song Ying-jie

(State Key Lab. of Integrated Service Networks, Xidian University, Xi'an 710071, China)

**Abstract:** Soft Output Viterbi Algorithm (SOVA) is more practical than MAP in the algorithms for decoding Turbo codes, because of its shorter decoding delay. In order to decrease further the decoding delay and increase the decoding speed, based on the simple analysis of the radix-4Max-Log-MAP algorithm, a novel radix-4SOVA algorithm is proposed and a comprehensive mathematical explanation is given. The key of the algorithm is to propose a novel reliability update method which achieves the reliability update after combining two-step state transitions in trellis diagram. Simulation results show that the BER performance of the novel algorithm is very close to that of radix-4Max-Log-MAP, and approaches radix-4MAP when the extrinsic information coefficient is adaptively added. Moreover, it reduces greatly the decoding delay and memory unit and achieves a good trade-off with the computational complexity.

**Key words:** Iterative decoding; Turbo codes; Convolutional Turbo Codes (CTC); Radix-4 Soft Output Viterbi Algorithm (Radix-4SOVA); MAP algorithm

### 1 引言

Turbo码以逼近Shannon极限的良好纠错性能成为了第三代(3G)移动通信系统标准的差错控制编码之一。随着通信技术对信息的实时性和传输速率的要求越来越高, 如何设计低延时、高吞吐率的Turbo译码技术, 是一个很有实际意义的工作。

Turbo 码的译码算法有两类: MAP<sup>[1]</sup>和SOVA<sup>[2]</sup>。MAP 根据所有可能路径计算软信息, 具有优异的译码性能, 但是需要前向和后向两次递归过程, 具有较大的译码延迟; 而一般意义的 SOVA 主要根据幸存路径来计算软信息, 译码时只需要在

前向递归过程中回溯一个固定长度, 因此译码延迟相对较小, 但是存在译码性能差的缺点。因此有不少文献对 SOVA 进行了改进, 提出了具有 MAP 类译码性能的 SOVA 算法<sup>[3-6]</sup>。但是对具有更低译码延迟的基于两步状态转移合并译码的基-4SOVA<sup>[7]</sup>, 目前还没有文献给出具有 MAP 类译码性能的数学推导。

本文正是基于上述情况, 提出了一种新的基-4SOVA 算法并做出了完整的数学推导, 新算法的误码率性能非常接近基-4Max-Log-MAP 算法, 当自适应引入外信息系数后, 逼近基-4MAP, 而且新算法具有译码延迟小、存储资源占用少等优点, 达到了与计算复杂度的良好折中。另外新算法还有一个独到之处, 它不仅适用于基于比特交织的二进制的 Turbo 码, 同时也适用于基于比特对交织的双二进

2010-12-16 收到, 2011-03-14 改回

长江学者和创新团队发展计划(IRT0852)和高等学校学科创新引智计划(B08038)资助课题

\*通信作者: 任德锋 rdf\_xidian@126.com

制的卷积 Turbo 码(CTC), 而传统的基-4SOVA 只适用于基于比特交织的二进制的 Turbo 码且译码性能较差。因此本文提出的具有次优译码性能的新基-4SOVA 译码算法具有一定的使用价值。

## 2 部分符号的定义

Turbo 码(或 CTC)译码器通过在两个分量译码器(分量译码器 1 和分量译码器 2)之间交换外信息进行迭代译码, 使信息比特(或比特对)的先验信息从无到有, 从有到越来越可靠, 这也是 Turbo 码(或 CTC)有如此优异译码性能的本质所在。由于两个分量译码器的工作原理完全相同, 因此笔者只给出对应分量译码器 1 的一些符号的定义, 来推导新的基-4SOVA 译码算法。

假设对应分量译码器 1 的分量码编码器的输出序列为  $\mathbf{x} = \mathbf{x}_1^N = (x_1, x_2, \dots, x_k, \dots, x_N)$ , 其中  $x_k = (x_k^s, x_k^p)$ ,  $x_k^s$  和  $x_k^p$  分别表示  $k$  时刻输出的系统信息和校验信息; 分量译码器 1 的输入序列为  $\mathbf{y} = \mathbf{y}_1^N = (y_1, y_2, \dots, y_k, \dots, y_N)$ , 其中  $y_k = (y_k^s, y_k^p)$ ,  $y_k^s$  和  $y_k^p$  分别表示  $k$  时刻接收到的系统信息和校验信息。传输信道模型为

$$\left. \begin{aligned} y_k^s &= a_k^s(2x_k^s - 1)\sqrt{E_s} + n_k^s \\ y_k^p &= a_k^p(2x_k^p - 1)\sqrt{E_s} + n_k^p \end{aligned} \right\} \quad (1)$$

其中  $a_k^s$  和  $a_k^p$  为信道衰落因子, 对于 AWGN 信道,  $a_k^s = a_k^p = 1$ ;  $n_k^s$  和  $n_k^p$  是两个独立同分布的高斯噪声值, 它们的均值为 0, 方差为  $\sigma^2 = N_0/2$ ,  $E_s$  是单个调制符号的能量。

令  $\mathbf{u} = (u_1, u_2, \dots, u_k, \dots, u_N)$  表示编码器输入的信息序列,  $u_{k,k+1} = (u_k, u_{k+1})$  ( $k$  为奇数)表示相邻两个时刻编码器输入的信息比特组成的一个信息比特对,  $L_a(u_{k,k+1} = AB)$  表示关于  $u_{k,k+1}$  的先验信息,  $L(u_{k,k+1} = AB | \mathbf{y}_1^N)$  表示关于  $u_{k,k+1}$  的对数似然比, 其中  $AB = 00, 01, 10, 11$ 。  $L_a(u_{k,k+1} = AB)$  和  $L(u_{k,k+1} = AB | \mathbf{y}_1^N)$  的定义如下:

$$L_a(u_{k,k+1} = AB) \triangleq \ln \frac{P(u_{k,k+1} = AB)}{P(u_{k,k+1} = 00)} \quad (2)$$

$$L(u_{k,k+1} = AB | \mathbf{y}_1^N) \triangleq \ln \frac{P(u_{k,k+1} = AB | \mathbf{y}_1^N)}{P(u_{k,k+1} = 00 | \mathbf{y}_1^N)} \quad (3)$$

显然,  $L_a(u_{k,k+1} = 00)$  和  $L(u_{k,k+1} = 00 | \mathbf{y}_1^N)$  都等于零。

## 3 基-4Max-Log-MAP 译码算法

基-4 算法的基本思路是将两步状态转移合并到一起译码, 因此一个时刻可以完成两比特的译码, 从而提高译码速度。由参考文献[8-10]给出的基-4Max-Log-MAP 算法可知, 比特对  $u_{k,k+1}$  的对数似

然比的计算公式为

$$\begin{aligned} L(u_{k,k+1} = AB | \mathbf{y}_1^N) &= \max_{(s'', s)} [\alpha_{k-1}(s'') + \gamma_{k,k+1}(s'', s) + \beta_{k+1}(s)] \\ &\quad - \max_{(s'', s)} [\alpha_{k-1}(s'') + \gamma_{k,k+1}(s'', s) + \beta_{k+1}(s)] \\ &\quad - \max_{(s'', s)} [\alpha_{k-1}(s'') + \gamma_{k,k+1}(s'', s) + \beta_{k+1}(s)] \\ &\quad - \max_{(s'', s)} [\alpha_{k-1}(s'') + \gamma_{k,k+1}(s'', s) + \beta_{k+1}(s)] \end{aligned} \quad (4)$$

式(4)中,  $s$  和  $s''$  表示分量编码器的状态,  $s, s'' = 0, 1, \dots, 2^m - 1$ ,  $m$  是分量编码器的寄存器单元数。 $\gamma_{k,k+1}(s'', s)$  表示从  $k-1$  时刻状态  $s''$  转移到  $k+1$  时刻状态  $s$  的两步状态转移对应的分支度量, 且计算公式为

$$\begin{aligned} \gamma_{k,k+1}(s'', s) &= \ln P(S_k = s', S_{k+1} = s, y_{k,k+1} | S_{k-1} = s'') \\ &= A_{k,k+1} + \frac{L_c}{2} (y_k^s \cdot \tilde{x}_k^s + y_k^p \cdot \tilde{x}_k^p \\ &\quad + y_{k+1}^s \cdot \tilde{x}_{k+1}^s + y_{k+1}^p \cdot \tilde{x}_{k+1}^p) + L_a(u_{k,k+1}) \end{aligned} \quad (5)$$

其中  $A_{k,k+1}$  是一个相对常数, 在对数似然比计算过程中会被消除, 所以不需要考虑;  $L_c = 4E_s/N_0$  为信道可信度量值, 仅依赖于 SNR;  $y_{k,k+1} = (y_k, y_{k+1})$  ( $k$  为奇数)表示相邻两个时刻接收到的信息组成的信息对;  $\tilde{x}_k^s = 2x_k^s - 1$ ,  $\tilde{x}_k^p = 2x_k^p - 1$ ,  $\tilde{x}_{k+1}^s = 2x_{k+1}^s - 1$ ,  $\tilde{x}_{k+1}^p = 2x_{k+1}^p - 1$  且  $\tilde{x}_k^s, \tilde{x}_k^p, \tilde{x}_{k+1}^s, \tilde{x}_{k+1}^p \in \{-1, 1\}$ ;  $L_a(u_{k,k+1})$  表示  $u_{k,k+1}$  的先验信息。

式(4)中,  $\alpha, \beta$  分别代表前向状态度量和后向状态度量, 其递归计算公式为

$$\alpha_{k+1}(s) = \max_{s''} [\alpha_{k-1}(s'') + \gamma_{k,k+1}(s'', s)] \quad (6)$$

$$\beta_{k-1}(s'') = \max_s [\beta_{k+1}(s) + \gamma_{k,k+1}(s'', s)] \quad (7)$$

用  $L_e(u_{k,k+1} = AB)$  表示  $u_{k,k+1}$  的外信息, 则其计算公式为

$$\begin{aligned} L_e(u_{k,k+1} = AB) &= L(u_{k,k+1} = AB | \mathbf{y}_1^N) - AL_c y_k^s \\ &\quad - BL_c y_{k+1}^s - L_a(u_{k,k+1} = AB) \end{aligned} \quad (8)$$

显然,  $L_e(u_{k,k+1} = 00)$  等于零。外信息  $L_e(u_{k,k+1} = AB)$  作为下一次分量译码的先验信息, 用来进行迭代译码, 这就是基-4Max-Log-MAP 译码算法的基本原理。

## 4 新基-4SOVA 译码算法

在上述基-4Max-Log-MAP 算法中由于存在后向状态度量  $\beta$  的计算, 产生了极大的译码时延, 本节的新算法用一种基于两步状态转移的可信度更新方法替代  $\beta$  的计算, 从而有效地降低了译码延迟。

### 4.1 计算累积路径度量

令状态序列  $\bar{\mathbf{s}}_{k+1}^s$  表示网格图上第  $k+1$  时刻状态  $S_{k+1} = s$  的幸存路径所走过的状态。  $\bar{\mathbf{s}}_{k+1}^s$  为正确路径的概率为

$$P(\bar{\mathbf{s}}_{k+1}^s | \mathbf{y}_1^{k+1}) = \frac{P(\bar{\mathbf{s}}_{k+1}^s, \mathbf{y}_1^{k+1})}{P(\mathbf{y}_1^{k+1})} \quad (9)$$

由于接收到序列  $\mathbf{y}_1^{k+1}$  的概率是一个常数，所以  $\bar{\mathbf{s}}_{k+1}^s$  为正确路径的概率与  $P(\bar{\mathbf{s}}_{k+1}^s, \mathbf{y}_1^{k+1})$  成正比，因此度量值只要使得  $P(\bar{\mathbf{s}}_{k+1}^s, \mathbf{y}_1^{k+1})$  最大化即可，考虑无记忆信道，令路径  $\bar{\mathbf{s}}_{k+1}^s$  前  $k-1$  个时刻的路径为  $\bar{\mathbf{s}}_{k-1}^{s''}$ ，则  $\bar{\mathbf{s}}_{k+1}^s$  的累积度量定义为

$$\begin{aligned} M(\bar{\mathbf{s}}_{k+1}^s) &\triangleq \ln P(\bar{\mathbf{s}}_{k+1}^s, \mathbf{y}_1^{k+1}) = \ln \{P(\bar{\mathbf{s}}_{k-1}^{s''}, \mathbf{y}_1^{k-1}) \\ &\quad \cdot P(S_k = s', S_{k+1} = s, y_{k,k+1} | S_{k-1} = s'')\} \\ &= \ln P(\bar{\mathbf{s}}_{k-1}^{s''}, \mathbf{y}_1^{k-1}) + \ln P(S_k = s', S_{k+1} = s, \\ &\quad y_{k,k+1} | S_{k-1} = s'') \\ &= M(\bar{\mathbf{s}}_{k-1}^{s''}) + \gamma_{k,k+1}(s'', s) \end{aligned} \quad (10)$$

此时的累积度量  $M(\bar{\mathbf{s}}_{k+1}^s)$  也可记为  $M(\bar{\mathbf{s}}_{k-1}^{s''} \rightarrow S_{k+1}^s)$ ，表示从第  $k-1$  时刻的幸存路径  $\bar{\mathbf{s}}_{k-1}^{s''}$  转移到第  $k+1$  时刻状态  $S_{k+1} = s$  的累积度量， $S_{k+1}^s$  表示第  $k+1$  时刻的状态  $s$ 。将分支度量计算式(5)代入式(10)就可求出下一时刻的累积度量。

#### 4.2 一种新的基于两步状态转移的可信度更新方法

和 MAP 类算法相比，SOVA 类只有前向度量，因此必须通过可信度更新来弥补缺少后向度量引起的性能损失。

二进制基-4 算法的栅格图把两步状态转移合并成一步，所以每个状态对应 4 个输入分支和 4 个输出分支，用  $\bar{\mathbf{s}}_{k-1}^{s_i}$  ( $i = 0, 1, 2, 3$ ) 表示转移到状态  $S_{k+1}^s$  在  $k-1$  时刻的 4 条幸存路径，用  $\delta$  (取偶数) 表示译码深度，则状态  $S_{k+1}^s$  关于  $u_{l,l+1} = (u_l, u_{l+1}) = AB$  ( $l = k-j, 2 \leq j \leq \delta$  且  $j$  取偶数， $l$  为奇数) 的概率为

$$\begin{aligned} P(u_{l,l+1} = AB, S_{k+1}^s, \mathbf{y}_1^{k+1}) &= \sum_{\bar{\mathbf{s}}_{k-1}^{s_i} \rightarrow S_{k+1}^s} P(u_{l,l+1} = AB, \bar{\mathbf{s}}_{k-1}^{s_i}, \mathbf{y}_1^{k-1}) \\ &\quad \cdot P(S_{k+1}^s, y_{k,k+1} | \bar{\mathbf{s}}_{k-1}^{s_i}) \\ &= \sum_{\bar{\mathbf{s}}_{k-1}^{s_i} \rightarrow S_{k+1}^s} P(u_{l,l+1} = AB | \bar{\mathbf{s}}_{k-1}^{s_i}, \mathbf{y}_1^{k-1}) \\ &\quad \cdot P(\bar{\mathbf{s}}_{k-1}^{s_i}, \mathbf{y}_1^{k-1}) P(S_{k+1}^s, y_{k,k+1} | \bar{\mathbf{s}}_{k-1}^{s_i}) \\ &= \sum_{\bar{\mathbf{s}}_{k-1}^{s_i} \rightarrow S_{k+1}^s} P(u_{l,l+1} = AB | \bar{\mathbf{s}}_{k-1}^{s_i}, \mathbf{y}_1^{k-1}) \\ &\quad \cdot P(\bar{\mathbf{s}}_{k-1}^{s_i}, \mathbf{y}_1^{k-1}, S_{k+1}^s, y_{k,k+1}) \\ &= \sum_{\bar{\mathbf{s}}_{k-1}^{s_i} \rightarrow S_{k+1}^s} P(u_{l,l+1} = AB | \bar{\mathbf{s}}_{k-1}^{s_i}, \mathbf{y}_1^{k-1}) \\ &\quad \cdot P(\bar{\mathbf{s}}_{k-1}^{s_i} | S_{k+1}^s, \mathbf{y}_1^{k+1}) P(S_{k+1}^s, \mathbf{y}_1^{k+1}) \end{aligned} \quad (11)$$

所以

$$\begin{aligned} P(u_{l,l+1} = AB | S_{k+1}^s, \mathbf{y}_1^{k+1}) &= \sum_{\bar{\mathbf{s}}_{k-1}^{s_i} \rightarrow S_{k+1}^s} P(u_{l,l+1} = AB | \bar{\mathbf{s}}_{k-1}^{s_i}, \mathbf{y}_1^{k-1}) \\ &\quad \cdot P(\bar{\mathbf{s}}_{k-1}^{s_i} | S_{k+1}^s, \mathbf{y}_1^{k+1}) \end{aligned} \quad (12)$$

式(12)两边取对数，可得基于两步状态转移的可信度更新公式。

$$\begin{aligned} \text{LLR}(u_{l,l+1} = AB | S_{k+1}^s, \mathbf{y}_1^{k+1}) &\triangleq \ln P(u_{l,l+1} = AB | S_{k+1}^s, \mathbf{y}_1^{k+1}) \\ &= \ln \sum_{\bar{\mathbf{s}}_{k-1}^{s_i} \rightarrow S_{k+1}^s} P(u_{l,l+1} = AB | \bar{\mathbf{s}}_{k-1}^{s_i}, \mathbf{y}_1^{k-1}) \\ &\quad \cdot P(\bar{\mathbf{s}}_{k-1}^{s_i} | S_{k+1}^s, \mathbf{y}_1^{k+1}) \\ &= \ln \sum_{\bar{\mathbf{s}}_{k-1}^{s_i} \rightarrow S_{k+1}^s} P(u_{l,l+1} = AB | \bar{\mathbf{s}}_{k-1}^{s_i}, \mathbf{y}_1^{k-1}) \\ &\quad \cdot \frac{P(\bar{\mathbf{s}}_{k-1}^{s_i}, S_{k+1}^s, \mathbf{y}_1^{k+1})}{P(S_{k+1}^s, \mathbf{y}_1^{k+1})} \\ &\approx \max_{\bar{\mathbf{s}}_{k-1}^{s_i} \rightarrow S_{k+1}^s} \{ \text{LLR}(u_{l,l+1} = AB | \bar{\mathbf{s}}_{k-1}^{s_i}, \mathbf{y}_1^{k-1}) \\ &\quad + M(\bar{\mathbf{s}}_{k-1}^{s_i} \rightarrow S_{k+1}^s) \\ &\quad - \max_{\bar{\mathbf{s}}_{k-1}^{s_j} \rightarrow S_{k+1}^s} \{ M(\bar{\mathbf{s}}_{k-1}^{s_j} \rightarrow S_{k+1}^s) \} \} \\ &= \max_{\bar{\mathbf{s}}_{k-1}^{s_i} \rightarrow S_{k+1}^s} \{ \text{LLR}(u_{l,l+1} = AB | \bar{\mathbf{s}}_{k-1}^{s_i}, \mathbf{y}_1^{k-1}) \\ &\quad + \Delta_{k-1 \rightarrow k+1}^{s_i \rightarrow s} \} \end{aligned} \quad (13)$$

其中  $\text{LLR}(u_{l,l+1} = AB | S_{k+1}^s, \mathbf{y}_1^{k+1})$  表示状态  $S_{k+1}^s$  关于  $u_{l,l+1} = AB$  的对数似然值，是一种可靠性度量；

$$\Delta_{k-1 \rightarrow k+1}^{s_i \rightarrow s} = M(\bar{\mathbf{s}}_{k-1}^{s_i} \rightarrow S_{k+1}^s) - \max_{\bar{\mathbf{s}}_{k-1}^{s_j} \rightarrow S_{k+1}^s} \{ M(\bar{\mathbf{s}}_{k-1}^{s_j} \rightarrow S_{k+1}^s) \}$$

表示从  $k-1$  时刻状态  $s_i$  转移到  $k+1$  时刻状态  $s$  的累积度量与转移到状态  $s$  的最大累积度量之差，可称为两步状态转移对应  $k+1$  时刻状态  $s$  的度量差；

定义  $\max^* \{ \}$  为： $\max^* \{ f(x_1), f(x_2), \dots, f(x_n) \} = \ln \left\{ \sum_{i=1}^n e^{f(x_i)} \right\}$ ，约等号“ $\approx$ ”两边由 Jacobi 对数近似公式： $\max^* \{ f(x_i) \} = \ln \left\{ \sum_i e^{f(x_i)} \right\} \approx \max \{ f(x_i) \}$  求得。

然后按式(14)进行赋值，则上述更新过程可以递归进行下去。

$$\left. \begin{aligned} \Delta_{k-1 \rightarrow k+1}^{u_{k,k+1}=AB \rightarrow s} &\rightarrow \text{LLR}(u_{k,k+1} = AB | S_{k+1}^s, \mathbf{y}_1^{k+1}) \\ \text{LLR}(u_{l,l+1} = AB | S_{k+1}^s, \mathbf{y}_1^{k+1}) &\rightarrow \\ &\text{LLR}(u_{l,l+1} = AB | \bar{\mathbf{s}}_{k+1}^s, \mathbf{y}_1^{k+1}) \end{aligned} \right\} \quad (14)$$

其中  $\Delta_{k-1 \rightarrow k+1}^{u_{k,k+1}=AB \rightarrow s}$  表示转移到  $k+1$  时刻状态  $s$  的 4 条路径中对应  $u_{k,k+1} = AB$  分支的累积度量与最大累积度量之差。

由可信度更新公式(13)，式(14)的更新过程可知，可信度更新时考虑了所有的路径，所以可以认为当译码深度足够大时(一般为约束长度的 5 倍)，比特对的可信度已经选择了全局最优路径上的软信息，因此采用新更新方法的新基-4SOVA 算法应该具有较好的译码性能。

### 4.3 计算对数似然比

假设译码深度  $\delta$  (为偶数) 足够大, 使得  $k+1$  时刻的全部幸存路径在  $k+1+\delta$  时刻已重合,  $S_{k+1+\delta}^{s^*}$  表示最大似然(ML)路径在  $k+1+\delta$  时刻的状态为  $s^*$ , 则由式(3), 式(13)可知  $u_{k,k+1} = AB$  的对数似然比可表示为

$$\begin{aligned} L(u_{k,k+1} = AB | \mathbf{y}_1^N) &= \text{LLR}(u_{k,k+1} = AB | S_{k+1+\delta}^{s^*}, \mathbf{y}_1^{k+1+\delta}) \\ &\quad - \text{LLR}(u_{k,k+1} = 00 | S_{k+1+\delta}^{s^*}, \mathbf{y}_1^{k+1+\delta}) \end{aligned} \quad (15)$$

将式(15)代入式(8)可求出外信息, 进而作为下一次分量译码所需的先验信息。由此译码可以不断地迭代进行。

### 4.4 新算法的使用范围

由于上述新算法每一步都是基于比特对推导的, 显然适用于基于比特对交织的双二进制的卷积 Turbo 码。

又由于,  $u_k$  的对数似然比:  $L(u_k | \mathbf{y}_1^N) = \ln \frac{P(u_k = 1 | \mathbf{y}_1^N)}{P(u_k = 0 | \mathbf{y}_1^N)}$

$$\begin{aligned} &= \ln \frac{P(u_{k,k+1} = 10 | \mathbf{y}_1^N) + P(u_{k,k+1} = 11 | \mathbf{y}_1^N)}{P(u_{k,k+1} = 00 | \mathbf{y}_1^N) + P(u_{k,k+1} = 01 | \mathbf{y}_1^N)} \\ &= \ln \frac{e^{L(u_{k,k+1}=10|\mathbf{y}_1^N)} + e^{L(u_{k,k+1}=11|\mathbf{y}_1^N)}}{e^{L(u_{k,k+1}=00|\mathbf{y}_1^N)} + e^{L(u_{k,k+1}=01|\mathbf{y}_1^N)}} \\ &= \max^* [L(u_{k,k+1} = 10 | \mathbf{y}_1^N), L(u_{k,k+1} = 11 | \mathbf{y}_1^N)] \\ &\quad - \max^* [L(u_{k,k+1} = 00 | \mathbf{y}_1^N), L(u_{k,k+1} = 01 | \mathbf{y}_1^N)] \end{aligned} \quad (16)$$

同理  $u_{k+1}$  的对数似然比:

$$\begin{aligned} L(u_{k+1} | \mathbf{y}_1^N) &= \max^* [L(u_{k,k+1} = 01 | \mathbf{y}_1^N), \\ &\quad L(u_{k,k+1} = 11 | \mathbf{y}_1^N)] \\ &\quad - \max^* [L(u_{k,k+1} = 00 | \mathbf{y}_1^N), \\ &\quad L(u_{k,k+1} = 10 | \mathbf{y}_1^N)] \end{aligned} \quad (17)$$

所以, 新算法也适用于基于比特交织的二进制的 Turbo 码。

另外, 新算法稍微改变也可用于卷积码与其它

码的级联译码、码间干扰(ISI)信道中的信号检测和迭代均衡等。因此本文提出的新基 4-SOVA 译码算法具有广泛的运用范围。

## 5 复杂度与仿真性能分析

### 5.1 复杂度比较

表 1 给出了分量码为(2,1,m)卷积码时, 4 种译码算法平均每步计算复杂度以及延迟复杂度和空间复杂度的比较, 其中  $\delta$  表示译码深度( $\delta$  取偶数, 单位: bit)。

由表 1 知, 基-4MAP 的计算复杂度显然高于基-4Max-Log-MAP, 而传统基-4SOVA 和新基-4SOVA 的计算复杂度与译码深度  $\delta$  有关。通常  $\delta$  取约束长度的 5 倍, 当  $m = 3$  时,  $\delta$  可取 20, 此时传统基-4SOVA 的计算复杂度最低, 新基-4SOVA 存在大量的加减法运算和最大(小)值操作, 其计算复杂度高于基-4Max-Log-MAP 和传统基-4SOVA, 但远低于基-4MAP。

在延迟方面, 基-4MAP 和基-4Max-Log-MAP 的延迟复杂度为  $O(N/2)$ , 所造成的时延对许多实际应用而言是不能接受的; 而新基-4SOVA 和传统基-4SOVA 的延迟复杂度为  $O(\delta/2)$ , 与  $N$  无关, 因此远小于基-4MAP 和基-4Max-Log-MAP。在空间复杂度方面, 基-4MAP 和基-4Max-Log-MAP 的空间复杂度为  $O(2 \times 2^m \times N/2)$ , 传统基-4SOVA 的空间复杂度为  $O(2 \times 2^m \times \delta + 2^m)$ , 而新基-4SOVA 的空间复杂度为  $O(3 \times 2^m \times \delta/2 + 2^m)$ , 显然传统基-4SOVA 和新基-4SOVA 所需的存储量空间相当, 并远小于基-4MAP 和基-4Max-Log-MAP。

### 5.2 仿真性能分析

为了验证本文提出的新基-4SOVA 译码算法的译码性能, 本文进行了计算机上的仿真模拟。仿真条件如下: 采用典型的编码速率为 1/3 的 Turbo 编码器, 分量编码器的存储级数为  $m = 3$ , 生成多项式为(13,15), 交织器选用 LTE 标准<sup>[11]</sup>中的二次置换多项式交织器, 调制方式为 QPSK, 迭代次数为 8 次, 噪声环境为高斯白噪声信道。

表 1 分量码为(2,1,m)卷积码的 4 种译码算法的复杂度比较

译码算法	平均每步的计算复杂度				延迟复杂度	空间复杂度
	加减法运算	乘法运算	最大(小)值操作	指数(对数)运算		
基-4MAP	$10 \times 2^m + 36$	$16 \times 2^m + 4$	0	19	$O(N/2)$	$O(2 \times 2^m \times (N/2))$
基-4Max-Log-MAP	$16 \times 2^m + 39$	4	$10 \times 2^m - 4$	0	$O(N/2)$	$O(2 \times 2^m \times (N/2))$
传统基-4SOVA	$6 \times 2^m + 40$	4	$3 \times 2^m + (\delta + 1)$	0	$O(\delta/2)$	$O(2 \times 2^m \times \delta + 2^m)$
新基-4SOVA	$7 \times 2^m + 43 + \delta \times 8 \times 2^m$	4	$3 \times 2^m + \delta \times 6 \times 2^m$	0	$O(\delta/2)$	$O(3 \times 2^m \times (\delta/2) + 2^m)$

图1给出了交织长度  $N = 2880$  bit时, 译码深度  $\delta$  对新基-4SOVA算法的误码率性能的影响, 同时图中还给出了在相同仿真条件下采用基-4Max-Log-MAP算法的误码率曲线。由图可以看到, 当  $\delta \geq 20$  时, 新基-4SOVA算法的误码率性能已经非常接近基-4Max-Log-MAP算法;  $\delta = 20$  与  $\delta = 50$  相比, 新基-4SOVA算法在误码率  $BER = 10^{-5}$  处性能仅下降了  $0.02$  dB, 可以忽略。因此  $\delta = 20$  时, 新基-4SOVA算法的误码率已经基本收敛。

图2给出了更新深度  $\delta = 20$  bit, 交织长度  $N = 48, 960, 2880$  和  $4800$  bit时, 新基-4SOVA算法与基-4Max-Log-MAP算法的性能比较。由图可以看到, 新基-4SOVA算法的误码率性能非常接近基-4Max-Log-MAP算法。新算法有如此良好性能的原因在于: 新基-4SOVA算法在用基于两步状态转移的可信度更新公式(13), 式(14)进行可信度更新时没有舍弃任何路径, 所以利用式(15)计算对数似然比时, 选择了最佳的4条路径, 因此性能非常接近基-4Max-Log-MAP算法。

SOVA算法比MAP算法输出的软判决值偏大, 或者说对判决可信度的估计过高, 致使性能下降。本文通过给式(8)的外信息乘以一个小于1的外信息系数(EIC)来减小可信度的偏差。图3给出了更新深度  $\delta = 20$  bit且交织长度  $N = 2880$  bit, 在信噪比为

$0.4$  dB和 $0.7$  dB时, EIC从0.6到1取值采用新基-4SOVA算法所得的误码率曲线。由图可知, EIC的引入可以改善误码率性能, 当信噪比为 $0.4$  dB时, EIC取 $0.7$ 性能最好, 当信噪比为 $0.7$  dB时, EIC取 $0.75$ 性能最好。

通过类似的仿真可以得到在不同信噪比下的EIC的最佳值如表2。由表2的数据可知随着信噪比的增大, EIC的最佳值呈现向1逐渐逼近的趋势。

表2 不同信噪比时EIC最佳值

$E_b/N_0$ (dB)	0	0.2	0.4	0.6	0.8	1
EIC最佳值	0.65	0.7	0.7	0.75	0.85	0.9

图4给出了更新深度  $\delta = 20$  bit且交织长度  $N = 2880$  bit, 自适应引入EIC最佳值的新基-4SOVA与未引入EIC的新基-4SOVA, 传统基-4SOVA, 基-4MAP, 基-4Max-Log-MAP的性能比较。由图可知, 自适应地引入EIC最佳值后, 新基-4SOVA的误码率性能与未引入EIC时相比又获得了进一步的改善, 与基-4MAP相比在误码率  $BER = 10^{-5}$  处性能仅下降了  $0.1$  dB, 而与传统基-4SOVA相比在误码率  $BER = 10^{-5}$  处性能提高  $0.7$  dB。因此我们极力推荐在实用时采用自适应引入EIC最佳值的新基-4SOVA。

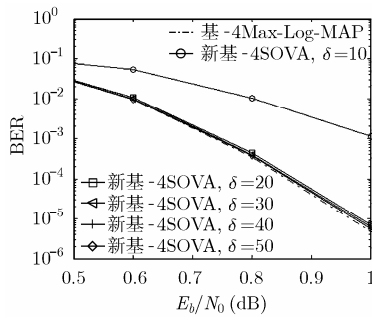


图1 交织长度为2880时译码深度对译码性能的影响

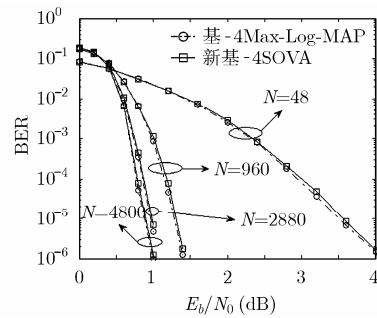


图2 两种算法的误码率性能比较

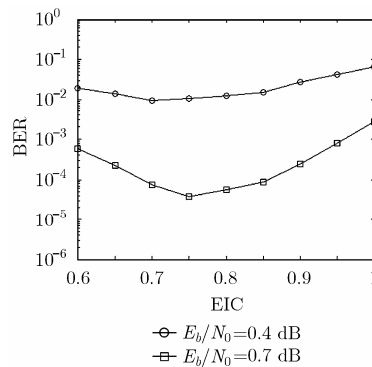


图3 不同EIC值的误码率曲线

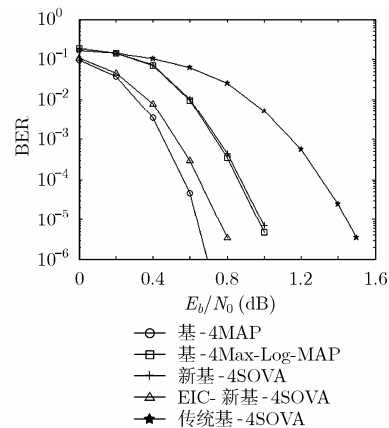


图4 自适应引入EIC最佳值的新基-4SOVA与其它4种算法的性能比较

## 6 结束语

本文提出了一种新的基-4SOVA 算法并做出了完整的数学推导,新算法具有译码性能优良、译码延迟小、存储资源占用少等优点,达到了与计算复杂度的良好折中。通过使用范围的分析知,它不仅适用于基于比特交织的二进制的 Turbo 码,同时也适用于基于比特对交织的双二进制的卷积 Turbo 码。另外,新算法还具有一般意义,稍做改变可用于卷积码与其它码的级联译码、码间干扰信道中的信号检测和迭代均衡等,因此本文提出的新基-4SOVA 译码算法具有一定的使用价值。

### 参考文献

- [1] Berrou C, Glavieux A, and Thitimajshima P. Near Shannon limit error-correcting coding and decoding: Turbo-codes.1[C]. Proceedings of IEEE International Conference on Communications 1993, Geneva, Switzerland, May 23-26, 1993, 2: 1064-1070.
- [2] Berrou C, Adde P, Angui E, *et al.* A low complexity soft-output viterbi decoder architecture[C]. Proceedings of IEEE International Conference on Communications 1993, Geneva, Switzerland, May 23-26, 1993, 2: 737-740.
- [3] Lang Lin and Cheng R S. Improvements in SOVA-based decoding for turbo codes[C]. Proceedings of IEEE International Conference on Communications 1997, Montreal, Canada, June 8-12, 1997, 3: 1473-1478.
- [4] Jun Tan and Stuber G L. Soft Output Viterbi Algorithm (SOVA) for non-binary Turbo codes[C]. Proceedings of IEEE International Symposium on Information Theory 2000, Sorrento, Italy, June 25-30, 2000: 483.
- [5] Liu J J and Tu G F. Iterative decoding of non-binary Turbo codes using symbol based SOVA algorithm[C]. Proceedings of IEEE International Conference on Communications, Circuits and Systems 2006, Guilin, China, June 25-28, 2006, 2: 689-693.
- [6] Ang Lay-hong, Lim Wee-guan, and Kamuf M. Modification of SOVA-based algorithms for efficient hardware implementation[C]. Proceedings of IEEE 71st Vehicular Technology Conference, Taipei, May 16-19, 2010: 1-5.
- [7] Haratsch E F and Fitzpatrick K K. A Radix-4 soft-output viterbi architecture[C]. IEEE International Symposium on VLSI Design, Automation and Test, Hsinchu, Apr. 23-25, 2008: 224-227.
- [8] Chuang Hsiang-tsung, Tseng Kai-hsin, and Fang Wai-chi. A high-throughput Radix-4 Log-MAP decoder with low complexity LLR architecture[C]. IEEE International Symposium on VLSI Design, Automation and Test, Hsinchu, Apr. 28-30, 2009: 231-234.
- [9] Lin C H, Chen C Y, Wu A Y, *et al.* Low-power memory-reduced traceback MAP decoding for double-binary convolutional Turbo decoder[J]. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2009, 56(5): 1005-1016.
- [10] 张成, 苏文艳, 刘亮, 等. 具有高速递归结构的基-4MAP 译码器[J]. *计算机工程与应用*, 2009, 45(3): 77-80.  
Zhang Cheng, Su Wen-yan, Liu Liang, *et al.* Radix-4 MAP decoder with fast recursion architecture[J]. *Computer Engineering and Applications*, 2009, 45(3): 77-80.
- [11] 3rd Generation Partnership Project. 3G TS 36.212 v8.4.0. Multiplexing and channel coding (Release 8)[S]. Sep. 2008.

任德锋: 男, 1983年生, 博士生, 研究方向为信道编码、认知无线电.

葛建华: 男, 1961年生, 教授, 博士生导师, 研究方向为无线通信、高清晰度数字电视技术等.

王 勇: 男, 1976年生, 副教授, 硕士生导师, 研究方向为无线宽带多媒体通信等.