Turbo 码的一种高效改进型 MAP 译码算法

赵宏宇　　范平志
(西南交通大学移动通信研究所信息编码与传输省重点实验室　成都　610031)

摘　要：该文给出了一种改进型最大后验概率(MAP)译码算法用于实现并行级联卷积码(Turbo 码)的最优译码。与基于对数域的 Log-MAP 算法相比较，该文给出的算法不引入对数域，但能够完全消除标准 MAP 算法在迭代过程中必须进行的大量指数和对数运算。计算机仿真结果表明，这种具有最优纠错性能的改进型 MAP 算法能够显著减少运行时间，其译码效率甚至优于牺牲了较多纠错性能的最快速的对数域 MAP 译码算法(Max-Log-MAP)。

关键词：Turbo 码；最大后验概率译码；译码时延
中图分类号：TN911.22　　**文献标识码**：A　　**文章编号**：1009-5896(2008)10-2397-05

An Efficient Improved MAP Decoding Algorithm for Turbo Codes

Zhao Hong-yu　　Fan Ping-zhi
(*Provincial Key Lab of Information Coding & Transmission*, *Institute of Mobile Communication*, *Southwest Jiaotong University*, *Chengdu* 610031, *China*)

Abstract: This paper presents an improved maximum *a posteriori* probability (MAP) decoding algorithm for optimal decoding of parallel concatenated convolutional codes (Turbo codes). Compared with the Log-MAP algorithms based on logarithm domain, the proposed algorithm does not introduce logarithm domain, but it can eliminate large amount of exponential and logarithm arithmetic operations required with the standard MAP algorithm in the iterative decoding process. Simulations demonstrate that he proposed algorithm, with optimal error correction performance, can significantly reduce the running time, such that its decoding efficiency is even slightly superior to that of the fastest logarithm domain MAP algorithm(Max-Log-MAP) which sacrifices considerable error correction performance.

Key words: Turbo code; MAP decoding; Decoding delay

1　引言

并行级联卷积码(Turbo 码)以接近 Shannon 极限的良好纠错性能[1]成为了第三代(3G)移动通信系统标准[2]推荐的差错控制编码之一。虽然 Turbo 码从发明至今已有十余年，但它的纠错性能却仍然不能用理论方法进行精确计算而必需通过大量的计算机仿真来估计。然而，Turbo 码采用的迭代译码算法具有相当大的计算复杂度，这使得大规模的译码仿真成为一项十分艰巨的任务。比如，当设计性能良好的交织器时，必须重点考察 Turbo 码在错误平层(error-floor)区域的纠错性能。在错误平层区域，Turbo 码译码产生的信息比特误码率和误帧率一般在 10^{-6}~10^{-8} 以及 10^{-4}~10^{-6} 范围，相应的计算机译码程序往往需要耗费几天至数星期的时间才能完成几个样本点的误码率以及误帧率的估计。因此，研究和探求一种高效率的 Turbo 译码算法具有重要的现实意义。

目前公开文献发表的 Turbo 码译码算法主要有三类，一类是 Turbo 码发明者 Berrou 最早给出的改进型 BCJR 迭代算法[1]，这是一种最大后验概率(MAP)译码算法，一般称为标准 MAP 算法；第二类是基于对数域的 BCJR 迭代算法，即 Log-MAP 算法[3−6]；第三类是改进型 Viterbi 算法[6]。在这些算法中最常用的是 Log-MAP 算法，这是因为该算法要求的数值动态范围相对较小并且很容易用定点数来实现[7]。具有较高译码效率的 Log-MAP 算法又分为查表 Log-MAP[3]，线性近似 Log-MAP[4, 5]和 Max-Log-MAP[3]三种。这三种 Log-MAP 算法不要求任何指数和对数运算，甚至乘法和除法运算也由于对数域的引入而转化为加法和减法运算，因此译码效率比标准 MAP 算法有了显著提高。从这些算法的差错控制性能来看，标准 MAP 算法的纠错性能最佳，称为最优译码；查表和线性近似 Log-MAP 算法具有非常接近最优译码的纠错性能，而 Max-Log-MAP 算法则以较大的纠错性能为代价，换得了译码效率的进一步提高；改进型 Viterbi 算法尽管效率最高但纠错性能不理想，因而很少实际采用。国外有学者在最近几年还提出了 Log-MAP 与 Max-Log-MAP 混合使用的算法[8, 9]，其目的是对译码效率与纠错

2007-03-19 收到，2007-08-22 改回
国家自然科学基金 (90604035) 及 NSFC/RFBR 研究项目 (60611120018)资助课题

性能进行不同程度的折中。

本文在深入研究以上 Turbo 码译码算法的基础上，重新推导得到了一种新的基于标准 MAP 算法的改进型 MAP 算法，这种算法没有引入对数域而是通过数学技巧来消除前向和后向迭代过程所需的全部指数和对数运算，从而极大地提高了计算机译码程序的效率。这种改进型 MAP 算法在理论上没有引入可能造成纠错性能减低的因素，因而属于最优译码算法的范畴。

2 标准 MAP 迭代译码算法

经典的Turbo码译码器由两个相同的1/2码率卷积码译码器组成，分别称为分量码译码器 0 和分量码译码器 1，如图 1 所示。

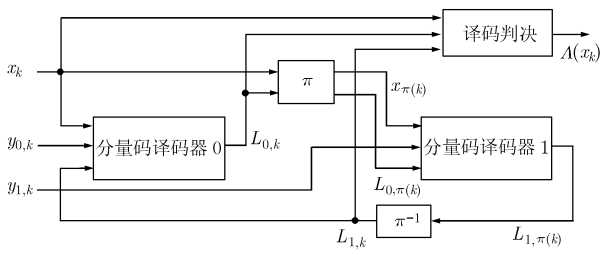

图 1 并行级联卷积码译码器原理图

串行执行的标准MAP迭代译码算法的一次迭代运算过程可以简单描述为：分量码译码器 0 以序列 $\{x_k\}$，$\{y_{0,k}\}$ 和 $\{L_{1,k}\}$ 为输入完成一次迭代，输出外信息序列 $\{L_{0,k}\}$；分量码译码器 1 以交织后的序列 $\{x_{\pi(k)}\}$，$\{L_{0,\pi(k)}\}$ 以及 $\{y_{2,k}\}$ 为输入完成一次迭代，输出外信息序列 $\{L_{1,\pi(k)}\}$。这里，x_k 表示连续送入分量码 0 的编码器进行编码的第 k 个信息比特 (k=1,2,3,$\cdots$,N，N 表示一个码块中的信息比特数目，称为帧长)；$y_{0,k}$ 表示分量码 0 的编码器送入信息比特 x_k 后输出的校验比特；$y_{1,k}$ 为分量码 1 的编码器输入信息比特 $x_{\pi(k)}$ 后输出的校验比特；$\pi(k)$ 为下标集合 $\{1,2,\cdots,N\}$ 上的一个一一映射函数。译码器实际输入的比特序列 $\{x_k\}$，$\{y_{0,k}\}$ 和 $\{y_{1,k}\}$，是二进制移相键控(BPSK)解调器输出的受到加性白色高斯噪声(AWGN)干扰的数字信号。

在每个分量码译码器内部，完成一次迭代需要 3 个步骤，它们分别是前向迭代、后向迭代和外信息计算。下面给出这 3 个步骤在分量码译码器 0 中的计算公式。对于分量码译码器 1，只需将以下公式中的 $x_k, y_{0,k}, L_{1,k}$ 分别替换为 $x_{\pi(k)}$，$y_{1,k}$ 和 $L_{0,\pi(k)}$。

2.1 前向迭代

$$\left.\begin{aligned}&\alpha_0(s)=1 \text{ 当 } s=0;\ \text{否则, } \alpha_0(s)=0, \quad k=0\\&\alpha_k(s)=\frac{\sum\limits_{s'}\sum\limits_{i}\alpha_{k-1}(s')\gamma_k(i,s',s)}{\sum\limits_{s}\sum\limits_{s'}\sum\limits_{i}\alpha_{k-1}(s')\gamma_k(i,s',s)},\ k=1,2,\cdots,N\end{aligned}\right\} \tag{1}$$

其中 $i\in\{+1,-1\}$ 表示信息比特；s 和 s'都表示卷积码编码器寄存器的状态编号，若卷积码编码器寄存器的长度为 v 比特，则状态编号依次为 0,1,$\cdots$,$n-1$，其中，n 表示状态数，有 $n=2^v-1$；三元组 (i,s',s) 表示当寄存器状态为 s' 时，送入信息比特 i 后，寄存器的状态将转移到 s，类似地，(i, s, s')表示状态为 s 时，送入信息比特 i 后状态将转移到 s'。

$\gamma_k(i,s',s)$ 称为状态转移概率测度，其计算公式按文献[6]给出如下：

$$\gamma_k(i,s',s)=\exp\left[\frac{1}{2}i\left(L_cx_k+L_{1,k}\right)+\frac{1}{2}p_{s'}^{(i)}L_cy_{0,k}\right] \tag{2}$$

其中 $p_{s'}^{(i)}\in\{+1,-1\}$，它表示寄存器状态为 s' 时，送入信息比特 i 后，编码器输出的校验比特。L_c 称为信噪比参数。对于加性白色高斯噪声(AWGN)信道，$L_c=2\sqrt{E_s}/\sigma^2$，其中 E_s 表示每个信道比特的等效传输能量；σ=$N_0/2$，其中 N_0 表示 AWGN 的双边功率谱密度。对于非时间相关的衰落信道，可令 $L_c=2E(A)\sqrt{E_s}/\sigma^2$ [10]，这里的 $E(A)$表示衰落幅度的均值。

2.2 后向迭代

$$\left.\begin{aligned}&\beta_N(s)=1/N, \qquad\qquad k=N\\&\beta_k(s)=\frac{\sum\limits_{s'}\sum\limits_{i}\beta_{k+1}(s')\gamma_{k+1}(i,s,s')}{\sum\limits_{s}\sum\limits_{s'}\sum\limits_{i}\alpha_k(s')\gamma_{k+1}(i,s',s)},\ k=N-1,N-2,\cdots,1\end{aligned}\right\} \tag{3}$$

2.3 外信息计算

$$L_{0,k}=\log\frac{\sum\limits_{(+1,s',s)}\alpha_{k-1}(s')\exp\left[\frac{1}{2}p_{s'}^{(+1)}L_cy_{0,k}\right]\beta_k(s)}{\sum\limits_{(-1,s',s)}\alpha_{k-1}(s')\exp\left[\frac{1}{2}p_{s'}^{(-1)}L_cy_{0,k}\right]\beta_k(s)} \tag{4}$$

式中，$\log(x)$表示自然对数函数。

分量码译码器 0 完成一次迭代以后，分量码译码器 1 开始一次迭代，重复进行这个迭代过程若干次以后，两个分量码译码器输出的译码判决信息由下面的公式给出。

2.4 译码判决

$$\Lambda(x_k)=L_cx_k+L_{0,k}+L_{1,k} \tag{5}$$

判决的方法是，当 $\Lambda(x_k)\ge 0$ 时，$u_k=+1$，否则 $u_k=-1$。这里，u_k 表示未经调制的第 k 个原始信息比特。

从式(1)–式(4)不难看出，标准 MAP 译码算法需要在迭代过程中反复计算指数和对数函数，因此译码效率很低。原始的对数域 Log-MAP 译码算法在计算状态转移概率测度时实际上仍然需要指数和对数函数运算，但是通过求最大值以及查表或者线性近似的方法，Log-MAP 算法可以回避状态转移概率测度中的指数和对数函数，译码效率因此得到显著提高。当然，查表或者线性近似会引起一点纠错性能的损失。与 Log-MAP 算法不同，下面将要给出的一种改进型 MAP 算法采用另一种方法来回避标准 MAP 算法中的指数和对数函数运算且无需付出纠错性能损失的代价。

3 改进型 MAP 迭代译码算法

为了消除标准 MAP 算法中的指数和对数运算，我们首

先按照如下方法重新实现状态转概率测度的计算。

在式(2)中，信息比特 i 与校验比特 $p_{s'}^{(i)}$ 可以分别表示为 $2j-1$ 和 $2q_{s'}^{(j)}-1$。其中，当 $i=1$ 时，$j=1$；当 $i=-1$ 时，$j=0$。类似地，$p_{s'}^{(i)}=1$ 时，$q_{s'}^{(j)}=1$；$p_{s'}^{(i)}=-1$ 时，$q_{s'}^{(j)}=0$。于是，状态转移概率测度 $\gamma_k(i,s',s)$ 可以表示为符号 j 和 $q_{s'}^{(j)}$ 的函数，即

$$\gamma_k(j,s',s)=\exp\left[j\left(L_c x_k+L_{1,k}\right)+q_{s'}^{(j)}L_c y_{0,k}\right] \tag{6}$$

式(6)可以简单证明如下：

因为 $\gamma_k(i,s',s)\propto \Pr(x_k,y_{0,k}\mid i,s',s)\Pr(u_k=i)$，其中

$$\Pr(x_k,y_{0,k}\mid i,s',s)=\left\{\frac{1}{2\pi\sigma^2}\right\}^2 \cdot\exp\left\{-\frac{\left[x_k-(2j-1)\sqrt{E_s}\right]^2+\left[y_{0,k}-(2q_{s'}^{(j)}-1)\sqrt{E_s}\right]^2}{2\sigma^2}\right\}$$

$$=A_k\exp\left(jL_c x_k+q_{s'}^{(j)}L_c y_{0,k}\right)$$

$$\Pr(u_k=i)=\frac{\exp(jL_{1,k})}{1+\exp(L_{1,k})}=B_k\exp(jL_{1,k})$$

由于 A_k 和 B_k 是与三元组 (j,s',s) 无关的量，当它们代入前向、后向迭代式(1)和式(3)后，可以约分消除，式(6)因此得证。

采用式(6)的优点是可以消除状态转移概率测度以及外信息计算公式中的全部指数和对数函数，具体方法如下。

令 $\tilde{x}_k=\exp(L_c x_k)$，$\tilde{y}_{m,k}=\exp(L_c y_{m,k})$ 以及 $\tilde{L}_{m,k}=\exp(L_{m,k})$，其中 $m=0$ 或 1。用指数值 $\tilde{x}_k,\tilde{y}_{m,k},\tilde{L}_{m,k}$ 分别代替图 1 中的 $x_k,y_{m,k},L_{m,k}$ 作为两个分量码译码器的输入，于是式(6)的 4 个值(记为 v_0,v_1,v_2,v_3)只需要两次乘法运算就能全部得到，如表 1 所示。

表 1 式(6)的计算方法

$(j,q_{s'}^{(j)})$	$\gamma_k(j,s',s)$ 的 4 个值
(0, 0)	$v_0\leftarrow 1$
(0, 1)	$v_1\leftarrow \tilde{y}_{0,k}$
(1, 0)	$v_2\leftarrow \tilde{x}_k\tilde{L}_{1,k}$
(1, 1)	$v_3\leftarrow v_1v_2$

相应地，外信息计算式(4)将简化为

$$\tilde{L}_{0,k}=\frac{\sum_{(1,s',s)}\alpha_{k-1}(s')\exp\left[q_{s'}^{(1)}\log(\tilde{y}_{0,k})\right]\beta_k(s)}{\sum_{(0,s',s)}\alpha_{k-1}(s')\exp\left[q_{s'}^{(0)}\log(\tilde{y}_{0,k})\right]\beta_k(s)} \tag{7}$$

考虑到 $q_{s'}^{(j)}\in\{0,1\}$，可知 $\exp\left[q_{s'}^{(j)}\log(\tilde{y}_{0,k})\right]=1$ 或 $\tilde{y}_{0,k}$，因此式(7)实际上不需要指数和对数函数运算。

当用 $x_k,y_{m,k},L_{m,k}$ 作为输入时，分量码译码器 0 开始第一次迭代前所有 $L_{1,k}$ 的初值是 0。因此，当用 $\tilde{L}_{1,k}$ 代替 $L_{1,k}$ 以后，所有 $\tilde{L}_{1,k}$ 的初值应设置为 1。

最后，将译码判决式(5)修改为

$$\tilde{\Lambda}(x_k)=\tilde{x}_k\tilde{L}_{0,k}\tilde{L}_{1,k} \tag{8}$$

新的判决方法是，当 $\tilde{\Lambda}(x_k)\geq 1$ 时，$u_k=+1$，否则 $u_k=-1$。

依据式(6)–式(8)，我们得到了一种新的 MAP 算法，这种 MAP 算法的特点是用指数值 $\tilde{x}_k,\tilde{y}_{m,k},\tilde{L}_{m,k}$ 作为 Turbo 译码器的输入并且在迭代过程中不再需要进行任何指数和对数函数运算。

为了检验以上改进型 MAP 算法的纠错性能，给出计算机仿真结果，如图 2 所示。图 2 中，纵坐标 BER 表示比特误码率；横坐标 E_b/N_0 表示信噪比，其中 E_b 为每个信息比特的等效能量，而 N_0 是加性高斯噪声(AWGN)的双边功率谱密度。仿真中，采用 3GPP 标准推荐的 8 状态卷积码$(1, 15/13)_{\text{oct}}$ 作为分量码[2, 11]。该标准限定信息比特帧长 N 在 40bit~5114bit 之间。为了保证实验结果的可靠性，图 1 中的每条仿真曲线由 4 次独立的实验结果求平均得到。仿真程序由 Visual C++7.0 实现，所需要的大量随机数由 Visual C++7.0 调用 MATLAB7.01 的随机数发生器产生。从图 2 不难看出，本文提出的改进型 MAP 算法的纠错性能与标准 MAP 算法完全相当。

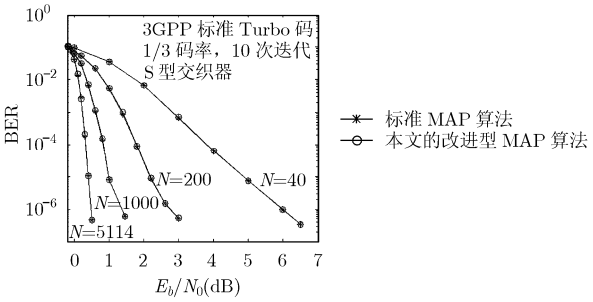

图 2 改进型 MAP 算法的纠错性能

4 多种 Turbo 译码算法的效率对比

为了考察以上改进型 MAP 算法的实际效率，本文将标准 MAP 算法、查表 Log-MAP 算法、线性近似 Log-MAP 算法、Max-Log-MAP 算法以及改进型 MAP 算法这 5 种算法的时间复杂度分析列于表 2。为了公平比较，所有算法均进行了尽可能的程序优化。除标准 MAP 算法以外，其它所有算法的状态转移率测度采用了相同的预算技术，即对第 k 步前向或后向迭代，在计算 n 个状态概率测度 $\alpha_k(s)$ 或 $\beta_k(s)$ 之前，事先计算好所需的 4 个状态转移概率测度值。对于本文提出的改进型 MAP 算法，这 4 个值只需要两次乘法运算(见表 1)，而对于 3 种 Log-MAP 算法，相应的 4 个值只需要两次加法运算。

对于查表 Log-MAP 算法，我们采用类似定点数译码的一种方案[7]来实现查表，这就是将待检索的关键字直接线性映射为线性表的下标(无需使用循环语句来检索)，从而最大限度地提高了查表速度。

表 3 给出了 5 种译码程序统计得到的平均每帧译码时

表 2 一个分量码译码器的一次迭代需要的数学运算次数对比

译码算法	加(减)法	乘(除)法	指数(对数)	求绝对值	求最大值	取整
标准 MAP	$15nN$	$(22n+1)N$	$(6n+1)N$	0	0	0
查表 Log-MAP	$(12n+3)N$	$(4n-2)N$	0	$(4n-2)N$	$(4n-2)N$	$(4n-2)N$
线性近似 Log-MAP	$(16n+1)N$	$(4n-2)N$	0	$(4n-2)N$	$(8n-4)N$	0
Max-Log-MAP	$(8n+5)N$	0	0	0	$(4n-2)N$	0
本文的改进型 MAP	$7nN$	$(10n+6)N$	0	0	0	0

表 3 不同计算机上的译码时间对比

译码算法	译码时间(毫秒/帧)		
	Pentium III 667M	Pentium-M 1.73G	Pentium 4 2.93G
标准 MAP	336.3	93.9	98.5
查表 Log-MAP	87.6	22.8	32.5
线性近似 Log-MAP	51.7	12.2	16.5
Max-Log-MAP	33.6	7.5	9.5
本文的改进型 MAP	27.2	5.7	5.6

间。表 3 的每项数据用连续译码 10^4 帧消耗的总时间求平均值得到。仿真条件是：帧长 N=1000，码率 1/3，3GPP 推荐的 8 状态分量码，E_b/N_0=1.8dB，S 型交织器，10 次迭代，AWGN 信道。程序运行总时间统计中，包括了产生随机数、编码、加噪声等操作的时间消耗。

表 3 清晰地表明，本文提出的改进型 MAP 算法的平均每帧译码时间仅相当于标准 MAP 算法的 6%左右，而查表 Log-MAP 算法和线性近似 Log-MAP 算法的平均每帧译码时间分别相当于标准 MAP 算法的 24%和 13%左右。此外，表 3 还说明本文的改进型 MAP 算法在时间效率上已经优于 Max-Log-MAP 算法。

推广使用本文提出的改进型 MAP 算法，现实意义十分明显。以笔记本电脑的 Pentium-M 1.73G 处理器为例，在表 3 的仿真条件下，完成一次错误平层区域的 10^{10} 个信息比特(1 千万帧)的译码，大约需要 16h，而用线性近似 Log-MAP 和查表 Log-MAP，同样的工作大约分别需要 34h 和 63h。

5 结束语

本文给出了一种新的改进型 MAP 算法。该算法是一种不牺牲纠错性能的非对数域最优译码算法。在多种计算机上的仿真程序实测表明，该算法的译码效率略优于牺牲了较多纠错性能的 Max-Log-MAP 算法而远远优于最主要的两种近似最优的译码算法，即查表 Log-MAP 和线性近似 Log-MAP。

从算法的时间复杂度分析(表 2)来看，本文的改进型 MAP 算法在乘(除)运算上虽逊于 Max-Log-MAP 算法，但实际达到的译码效率却略优于 Max-Log-MAP 算法。我们分析推测，其原因应该归功于 Intel 处理器的强大浮点运算能力。本文的改进型 MAP 算法，尽管在迭代译码之前仍然需要进行一趟指数运算，但在最消耗时间的迭代过程中没有任何比较大小运算(求最大值)、求绝对值运算以及浮点数化为整数(取整)运算，因此实际达到的时间效率反而更高。从表 2 中还可以注意到，线性近似 Log-MAP 应该比查表法 Log-MAP 运算时间复杂度差一些，但它的实际运行效率却好于查表法 70%~80%。此外，对比 Pentium-M 1.73G 与 Pentium 4 2.93G 处理器，不难发现，二者的浮点数四则运算(加、减、乘、除)能力基本相当，而其它运算(求最大值、求绝对值和取整)的能力则前者比后者明显高出一截。总之，这些实验结果说明，对于通用计算机上实现的 Turbo 码译码程序并不能单独依靠传统的时间复杂度分析来考查它们的实际运行效率。

虽然 Log-MAP 算法数值动态范围小能够很容易采用定点数来实现，但是在通用计算机上，采用定点数运算实际上需要附加更多的程序操作，如数值规范化、饱和限幅等，所以实际能够达到的仿真效率只会更低。本文提出的改进型 MAP 算法非常适合于 Turbo 码纠错性能计算机仿真，它能够极大地提高仿真的效率。

参 考 文 献

[1] Berrou C, Glavieux A, and Thitimajshima P. Near Shannon limit error-correcting coding and decoding. Proc. of IEEE Int. Conf. on Communications 1993, Geneva, 1993: 1064–1070.

[2] 3rd Generation Partnership Project, Multiplexing and channel coding(FDD), 3G TS 25.212, June 1999.

[3] Robertson P, Villebrun E, and Hoecher P. A comparison of optimal and sub-optimal MAP decoding algorithms operation in the Log Domain. Proc. of Int. Conf. on Communications 1995, Seattle, Gateway to Globalization, 1995: 1009–1013.

[4] Cheng J-F and Ottosson T. Linearly approximated Log-MAP algorithms for turbo decoding. Vehicular Tech. Conf. Proceedings 2000, Tokyo, 2000, 3: 2252–2256.

[5] Valenti M C. An efficient software radio implementation of the UMTS turbo code, Proc. of 2001 12th IEEE International Symposium on Personal, Indoor and Mobile

Radio Comm. 2001, 2: G-108-G-113.

[6] Offer H E and Papke L. Iterative decoding of binary block and convolutional codes. *IEEE Trans. on Inform. Theory*, 1996, 42(2): 429-445.

[7] Montorsi G and Benedetto S. Design of fixed-point iterative decoders for concatenated codes with interleavers. *IEEE Journal on Selected Areas in Comm.*, 2001, 19(Issue 5): 871-881.

[8] Park S J. Combined Max-Log-MAP and Log-MAP of turbo codes. *IEE Electronics Letters*, 2004, 40(4): 251-252.

[9] Papaharalabos S, Sweeney P, and Evans B G. SISO algorithm based on combined max/max* operations for turbo decoding. *IEE Electronics Letters*, 2005, 41(3): 142-143.

[10] Hall E K and Wilson S G. Design and analysis of turbo codes on Rayleigh fading channels. *IEEE Journal on Selected Areas in Comm.*, 1998, 16(Issue 2): 160-174.

[11] 王新梅，肖国镇. 纠错码——原理与方法(修订版). 西安：西安电子科技大学出版社，2001: 505-532.
Wang X M and Xiao G Z. Correcting Error Codes——Principles and Methods (Amended version), Xi'an: Xi'an Electronics Science and Technology University Press, 2001: 505-532.

赵宏宇： 男，1971年生，讲师，博士生，研究方向为信道编码理论与应用、包括迭代译码算法、交织器设计、信噪比估计等.

范平志： 男，1955年生，教授，博士生导师，研究方向为移动通信系统、包括编码理论与应用、序列设计、码分多址、协作通信等.