基于量化的 LDPC 译码算法的高效实现

马 卓* 杜栓义 王新梅
(西安电子科技大学 ISN 国家重点实验室 西安 710071)

摘 要：论文提出了一种采用 2 维折线逼近的和积译码算法实现方案，避免了使用与量化比特数成指数关系增长的查找表，降低了译码器的存储器消耗。基于上述方案提出了一种次小值修正的最小和算法。该算法通过 3 个 2 维折线逼近对最小值进行修正，获得了逼近浮点和积算法的译码性能。算法的修正过程只包含简单的算术和逻辑运算，便于 FPGA 实现。

关键词：信道编码；低密度奇偶校验码；和积算法；最小和算法；量化
中图分类号：TN911.22 **文献标识码**： A **文章编号**：1009-5896(2011)09-2273-05
DOI: 10.3724/SP.J.1146.2011.00041

Efficient Implementing of LDPC Decoding Algorithm Based on Quantization

Ma Zhuo Du Shuan-yi Wang Xin-mei
(*The State Key Lab of Integrated Service Networks*, *Xidian University*, *Xi'an* 710071, *China*)

Abstract: An implementing scheme of sum-product algorithm is proposed based on 2-dimation broken line approach, which avoids the look-up table with size related to the exponential of the number of quantization bits and reduces the memory consumption of the decoder. Then, an algorithm called second-minimum value corrected min-sum algorithm is proposed based on the implementing scheme proposed above. The algorithm use three 2-dimension piecewise approach to correct the min-sum algorithm and its performance is very close to that of the floating point sum-product algorithm. The correction process of this algorithm just includes simple arithmetic and logic operations, which is easy to be implemented by FPGA.

Key words: Channel coding; Low Density Parity-Check (LDPC) code; Sum-product algorithm; Min-sum algorithms; Quantization

1 引言

低密度奇偶校验码(LDPC)码是一类由稀疏的校验矩阵定义的分组纠错码，近年来受到了广泛的关注和研究。LDPC 码在采用和积译码算法时，可以取得非常接近于香农限的误码性能[1]，其复杂度只与校验矩阵中非零元素的个数相关，即具有相对于校验矩阵重量的线性复杂度。实际中通常采用查找表(LUT)的方式来实现和积算法中复杂函数的计算[2]。由于 LUT 的规模随着量化比特数的指数增加，因此和积译码译码器的量化精度往往受限于 LUT 的规模。最小和算法运算简单，具有较低的复杂度，其改进算法，包括归一化的最小和算法和偏移最小和算法在 FPGA 实现时被大量的采用[3]。但是改进的最小和算法在长码长和低码率时仍然与和积算法有较大的差距[4]，限制了其应用。

对于 LDPC 译码算法的改进，一方面可以在保证译码性能的前提下尽可能地降低和积算法的复杂度，另一方面可以对最小和算法进行更加精细的修正，以使其译码性能更加逼近和积算法。对于前者，文献[5–11]从不同角度给出了简化的方案。文献[5]针对核心函数 $\varphi(x)$ 可能取无限值所造成的数值稳定性问题提出了两种改进方案；文献[6]通过对校验矩阵的拆分以降低信息的传递次数从而降低和积译码算法的复杂度；文献[7]给出了两种适用于高码率情况下的基于大数逻辑的迭代译码算法；文献[8]对非均匀量化进行了优化使得译码器可以工作在 3–4 bit 的量化精度下。对于后者，文献[9]使用两个参数对最小值进行修正；文献[10,11]提出了一种基于校验节点度的分类修正的最小和算法。但是上述两种算法在长码长和低码率情况下的性能仍然有待改善。

本文通过分析和积译码算法中校验信息的更新运算，得到了一种基于量化的和积译码算法的高效实现方案，即 2 维折线逼近方案。然后基于该实现方案，提出了一种次小值修正的最小和算法。

2011-01-14 收到，2011-05-10 改回
国家 973 计划项目(2010CB328300)，国家自然科学基金(U0635003)和 111 基地基金(B08038)资助课题
*通信作者：马卓 zma@mail.xidian.edu.cn

2 和积算法与最小和算法

LDPC 码的置信传播译码算法可以用因子图上的消息传递过程来表示，和积算法与最小和算法的区别仅在于传递信息的计算方式不同。

设由变量节点 x_j 传递给校验节点 z_i 的信息记做变量信息 v_{ij}，由校验节点 z_i 传递给变量节点 x_j 的信息记做校验信息 u_{ij}，变量节点 x_j 的先验信息为 u_j。则和积译码的算法的信息更新规则如下：

$$u_j = 2y_j \big/ \sigma_n^2 \tag{1}$$

$$v_{ij} = u_j + \sum_{k\in M(j)\backslash i} u_{kj} \tag{2}$$

$$u_{ij} = 2\tanh^{-1}\left(\prod_{k\in N(i)\backslash j} \tanh\left(v_{ik}/2\right)\right) \tag{3}$$

其中 $M(j)$ 为与变量节点 x_j 相连的校验节点的集合，$N(i)$ 为与校验节点 z_i 相连的变量节点的集合。

最小和算法的初始信息计算和变量信息的更新规则与和积算法相同，其校验信息更新规则为

$$u_{ij} = \prod_{k\in N(i)\backslash j} \operatorname{sgn}(v_{ik}) \min_{k\in N(i)\backslash j}\left(\left|v_{ik}\right|\right) \tag{4}$$

3 和积算法的 2 维折线逼近实现

和积算法中，初始信息和变量信息的更新计算比较简单，实现时的难点主要在于校验信息的更新计算。对式(3)所示的和积算法的校验信息更新规则，可将其变换成如下的形式

$$u_{ij} = \left(\prod_{k\in N(i)\backslash j} \operatorname{sgn}(v_{ik})\right)\varphi^{-1}\left(\sum_{k\in N(i)\backslash j} \varphi\left(v_{ik}\right)\right) \tag{5}$$

$$\varphi(x) = \varphi^{-1}(x) = \log\left(\frac{e^{|x|}+1}{e^{|x|}-1}\right) = \log\left(\frac{1+e^{-|x|}}{1-e^{-|x|}}\right) \tag{6}$$

通常基于FPGA 实现的和积译码器大都通过查找表的方式实现函数 $\varphi(x)$。此时，量化比特数每增加一位，查找表的规模就增大一倍，因此在并行实现时查找表消耗的存储单元是非常可观的。此外，由于函数 $\varphi(x)$ 的输入和输出的动态范围相差非常大且函数值可能取无穷值($\varphi(0)=+\infty$)，从而造成了精度损失和数值稳定性的问题[5]，不利于提高量化译码器的性能。

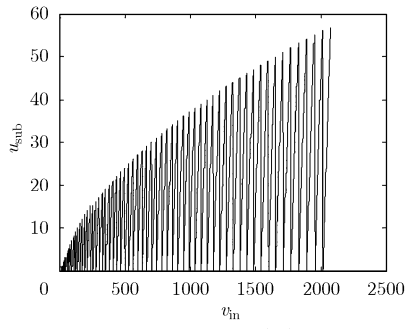
由于式(5)中的累加通常采用两两相加来实现[2]，本文将主要考察校验节点的串行更新公式

$$u_{\text{sub}} = \operatorname{sgn}(v_1)\operatorname{sgn}(v_2)\varphi^{-1}(\varphi(v_1)+\varphi(v_2)) \tag{7}$$

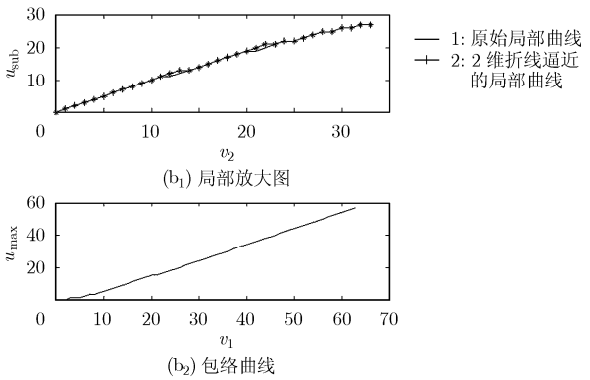
本文将式(7)中校验信息绝对值的更新过程 $\varphi^{-1}(\varphi(v_1)+\varphi(v_2))$ 看作一个二输入函数 $\psi(v_1,v_2)$，并对其进行折线逼近[5,12]。采用量化间隔为 0.125 的 7 bit 均匀量化方案(最高位为符号位)，可以画出函数 $\psi(v_1,v_2)$ 在 $0\le v_2\le v_1<2^6$ 的条件下，由不同输入对 (v_1,v_2)所得到的曲线，如图 1(a)所示，其横坐标为 $v_{\text{in}} = v_1(v_1+1)/2+v_2$。

可以看出函数 $\psi(v_1,v_2)$ 是按照 v_1 的取值分段的，且各段具有相似的形状，如图 $1(\text{b}_1)$中的曲线 1 所示。该曲线是 $v_1=34$ 时取出的 $\psi(v_1,v_2)$ 曲线的一段，由于此时 v_1 已经确定，横坐标可以用 v_2 来表示。在该局部曲线中，函数值 u_{sub} 在 $v_2=0$ 处取最小值 0，在 $v_2=v_1$ 处取最大值 27。$\psi(v_1,v_2)$ 的不同分段具有不同的最大值 $u_{\max}$，因此只要确定了 $u_{\max}$ 和 v_1 的的关系，即可确定 $\psi(v_1,v_2)$ 的形状。$u_{\max}$ 和 v_1 的关系可以由函数 $\psi(v_1,v_2)$ 的包络得到，该包络如图 $1(\text{b}_2)$中的曲线所示。

根据图 $1(\text{b}_2)$可以得到 $u_{\max}$ 和 v_1 的关系如表 1 所示。



(a) $\psi(v_1,v_2)$ 曲线



(b_1) 局部放大图

(b_2) 包络曲线

(b) 局部放大与包络

图 1 $\psi(v_1,v_2)$ 曲线及其局部放大图与包络

表 1 $u_{\max}$ 取值表

v_1	$0\le v_1\le 2$	$2<v_1\le 5$	$5<v_1\le 7$	$7<v_1\le 20$	$20<v_1\le 63$
$u_{\max}$	0	1	v_1-4	v_1-5	v_1-6

已知 $u_{\max}$ 即可对局部曲线进行逼近。逼近公式为

$$u_{\text{sub}}=\psi(v_1,v_2)=\begin{cases}\min(v_2>>\beta(v_{\text{mid}}),u_{\max}), & v_2\le v_{\text{low}}\\ v_2-1, & v_{\text{low}}<v_2\le v_{\text{mid}}\\ u_{\max}-((v_1-v_2)>>1)-1, & v_{\text{mid}}<v_2\le v_{\text{high}}\\ u_{\max}-(v_1-v_2)>>1, & v_{\text{high}}<v_2\le v_1\end{cases}\quad(8)$$

其中 v_{mid}，v_{low} 和 v_{higt} 为修正参数。v_{mid} 可由式(9)确定

$$v_{\text{mid}}=v_1-(v_1-u_{\max})<<1\quad(9)$$

v_{low} 和 v_{higt} 可由表 2 和表 3 得到。

函数 $\beta(v_{\text{mid}})$ 定义如下：

$$\beta(v_{\text{mid}})=\begin{cases}0, & v_{\text{mid}}>0\\ 1, & v_{\text{mid}}\le 0\end{cases}\quad(10)$$

表 2 v_{low} 取值表

v_1	$0\le v_1<10$	$v_1=10$	$11\le v_1<16$	$16\le v_1<20$	$20\le v_1<22$	$22\le v_1<25$	$25\le v_1\le 63$
v_{low}	63	0	1	2	3	4	v_1-20

表 3 v_{higt} 取值表

v_1	$0\le v_1<11$	$11\le v_1<20$	$20\le v_1<63$
v_{high}	0	v_1	v_1-6

式(8)所述的实现方案称为 2 维折线逼近方案。图 1(b_1)的曲线 2 所示为由式(8)得到的局部曲线。可以看出，本文给出的方案可以较好地逼近函数 $\psi(v_1,v_2)$，且只包含加法、比较和移位等简单运算，无需存储大规模的查找表，适合于 FPGA 实现。此外，本方案逼近的函数 $\psi(v_1,v_2)$ 的输入输出动态范围基本一致，避免了逼近函数 $\varphi(x)$ 所造成的精度损失和数值不稳定性问题。

4　次小值修正的最小和算法

在上一节中，我们将和积算法中变量信息的更新公式采用 2 维折线逼近的方法实现，降低了存储器的消耗，但是该方案的缺点是每计算一次 $\psi(v_1,v_2)$ 至少需要两个时钟的时延(一个时钟用来查表，一个时钟用来计算输出)。这样，在校验节点度数比较大时就会造成较大的总体时延，降低了译码器的工作效率。为此本文又提出了基于上述 2 维折线逼近方案的一种最小和算法的修正算法。

由上一节的分析可知，2 维折线逼近的过程实际上可以看作是由两个变量信息中绝对值较大的一个对绝对值较小的一个进行修正。基于上述修正过程，本文提出如下的简化方案：

$$u_{ij}=\begin{cases}\prod\limits_{k\in N(i)\setminus j}\operatorname{sgn}(v_{ik})\max(\psi(v_{ik_2},v_{ik_1})-1,0), & j\ne k_1,k_2\\ \prod\limits_{k\in N(i)\setminus j}\operatorname{sgn}(v_{ik})\max(\psi(v_{ik_3},v_{ik_1})-1,0), & j=k_2\\ \prod\limits_{k\in N(i)\setminus j}\operatorname{sgn}(v_{ik})\max(\psi(v_{ik_2},v_{ik_3})-1,0), & j=k_1\end{cases}\quad(11)$$

其中

$$v_{ik_1}\le v_{ik_2}\le v_{ik_3}\le v_{ik},\ k\in N(i)\setminus(k_1,k_2,k_3)\quad(12)$$

公式中的减 1 是对前述简化的补偿。

上述算法称为次小值修正的最小和算法。该算法在进行校验信息计算时，无论该校验节点的度为多少，只需要找到 3 个最小值并进行 3 次 2 维折线逼近即可，其计算复杂度和计算时延相对于和积算法大大降低。表 4 所示为对各种算法的计算复杂度的比较，比较基于度为 7 的校验节点，采用量化间隔为 0.125 的 7 bit 均匀量化方案。其中次小值修正的最小和算法的运算量包含了查找 3 个最小值所需要的运算量(针对度为 7 的校验节点)。实际上，采用次小值修正的最小和算法实现的 LDPC 译码器中，一个度为 7 的校验节点计算单元消耗的逻辑资源为 971，总的运算时延为 5 个时钟。

表 4 计算复杂度比较

	基于查表的和积算法实现	和积算法 2 维折线逼近方案	次小值修正的最小和算法
加法	0	$2d_c\times13=182$	$3\times13=39$
比较	0	$2d_c\times18=252$	$3\times18+15=69$
移位	0	$2d_c\times4=56$	$3\times4=12$
存储资源	$2d_c\times64\times6$ =5376 bit	0	0

5　仿真结果

为评估本文述及算法的性能，对其性能进行了仿真。仿真采用的调制方式为 BPSK，信道为加性高斯白噪声(AWGN)信道。设定译码器最大迭代次数为 50 次。作为对照，还对浮点的和积译码算法、量化的最小和算法、量化的 offset 最小和算法(offset=1)以及分类修正最小和算法进行了仿真。仿真中除浮点和积算法外，均采用量化间隔为 0.125 的 7 bit 均匀量化方案。

首先采用两种码率均为 $R=1/2$ 的准循环 LDPC(QC-LDPC)码进行了仿真，仿真结果如图 2(a)所示。其中实线所示为码 I，码 I 选用了 802.16e[13] 中规定的码长为 2304 的 LDPC 码；虚线所示为码 II，该码为基于围长最大准则构造的具有类似码 I 的准双对角线结构的 QC-LDPC 码，码长为 $L=23040$，扩张因子为 $z=720$，其度分布如下：

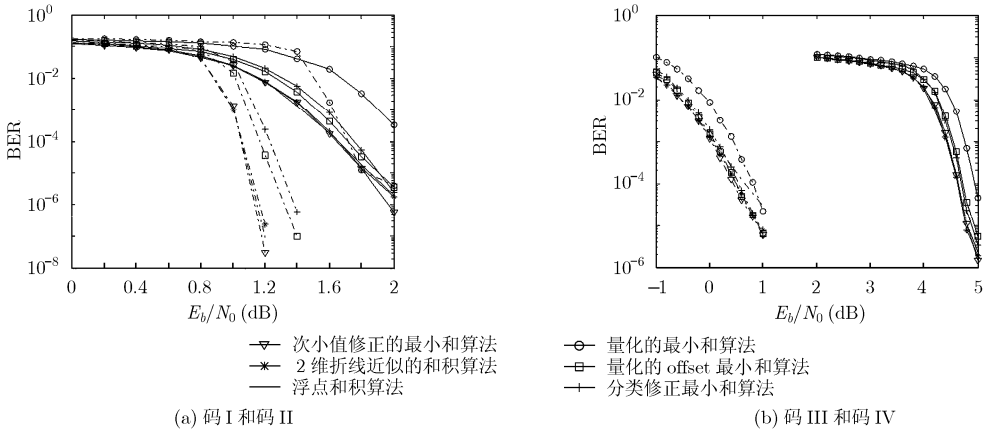
$$\left.\begin{aligned}\lambda(x)&=0.275229x+0.275229x^2+0.449542x^6\\ \rho(x)&=0.165138x^5+0.834862x^6\end{aligned}\right\} \quad (13)$$

由图 2 可以看出，本文所述的两种算法在两种码长下都具有逼近于浮点和积译码算法的性能，而最小和算法及其两种改进算法的性能与本文所述的算法有较大的差距，其中性能最好的 offset 最小和算法的性能也与次小值修正的最小和算法有 0.07 dB 到 0.15 dB 的差距。

然后对不同码率的 QC-LDPC 码进行了仿真。高码率的 LDPC 码(码 III，用实线表示)采用 802.16e 中码率为 $R=3/4$ 的码 A，码长设定为 2304。低码率的 LDPC 码(码 IV，用虚线表示)由前述的码 I 删掉校验矩阵的前 6 列得到，其码率为 $R=1/3$，码长设定为 768。仿真结果如图 2(b)所示。可以看出，本文提出的两种算法仍然具有较好的误码性能，特别是次小值修正的最小和算法在码 IV 的情况下其误码性能在整个信噪比区间上都优于浮点和积算法。

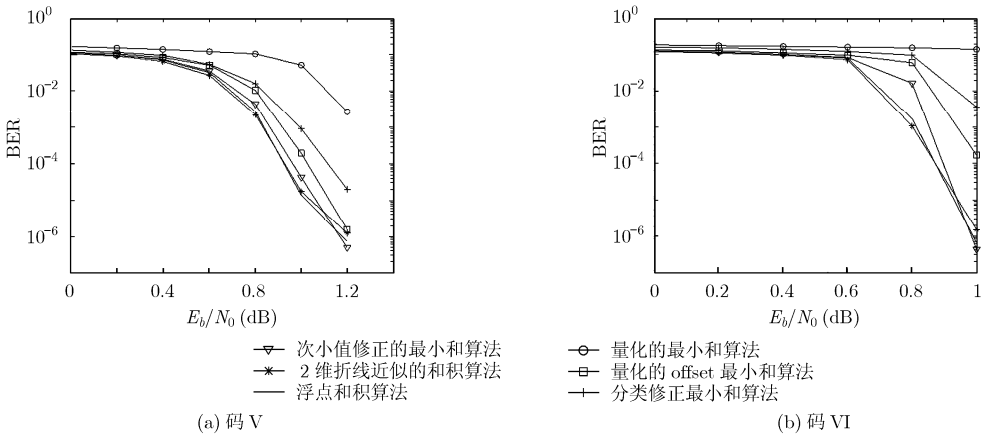
我们还采用 DVB-S2 规定的两种码长分别为 64800(码 V)和 16200(码 VI)的 LDPC 码进行了仿真，实际码率分别选为 1/2 和 4/9，仿真结果如图 3 所示。可以看出，2 维折线逼近的和积算法在这种情况下仍然具有逼近于浮点和积算法的性能，而次小值修正的最小和算法的性能相对于 QC-LDCP 码的情况略有下降，其与浮点和积算法的性能的差距最大达到了 0.1 dB 左右。尽管如此，该算法的性能仍然好于 offset 最小和算法和分类修正的最小和算法。

最后，我们采用码 II，并将扩张因子设为 256，在 FPGA 上实现了基于次小值修正的最小和算法的 LDPC 译码器，使用的 FPGA 为 Altera 公司的 EP3C50F484C8。综合结果如表 5 所示。



(a) 码 I 和码 II　　(b) 码 III 和码 IV

图 2 不同码长和码率 QC-LDPC 码译码性能比较



(a) 码 V　　(b) 码 VI

图 3 DVB-S2 LDPC 码译码性能比较

表 5 综合结果

逻辑资源	存储资源	最高时钟
42352	448000 bit	128 MHz

设定最大迭代次数为 50。每次迭代需要消耗的时钟数为 2×256+7=519，其中 7 为排空流水线所需要的额外时钟。此外，输入数据存入 RAM 也需要消耗 256 个时钟。则由最高时钟估算得到的吞吐率为 4096×128/(519×50+256)=20 Mbps。需要说明的是，本文实现所采用的 FPGA 的速度等级较低，如果采用更高速度等级的 FPGA，采用该算法将能够实现具有更高吞吐率的译码器。

6　结论

本文提出了两种基于量化的 LDPC 译码算法的实现方案，并通过仿真验证了两种算法在各种码率和码长的情况下都具有接近甚至好于浮点和积算法的性能，但是其复杂度却大大降低，是一种适于 FPGA 实现的量化的 LDPC 译码算法。

参 考 文 献

[1] MacKay D J C and Neal R M. Near shannon limit performance of low density parity check codes [J]. *Electronics Letters*, 1996, 32(18): 1645–1646.

[2] 张靖琳, 刘荣科, 赵岭. 高码率 LDPC 码译码器的优化设计与实现[J]. 电子与信息学报, 2009, 31(1): 83–86.
Zhang Jing-lin, Liu Rong-ke, and Zhao Ling. Optimized decoder design and implement for high rate LDPC codes [J]. *Journal of Electronics & Information Technology*, 2009, 31(1): 83–86.

[3] Jiang Nan, Peng Kewu, Song Jian, *et al*.. High- throughput QC-LDPC decoders [J]. *IEEE Transactions on Broadcasting*, 2009, 55(2): 251–259.

[4] Zhao J, Zarkeshvari F, and Banihashemi A H. On implementation of min-sum algorithm and its modifications for decoding Low-Density Parity-Check (LDPC) codes [J]. *IEEE Transactions on Communications*, 2005, 53(4): 549–554.

[5] Masera G, Quaglio F, and Vacca F. Finite precision implementation of LDPC decoders [C]. *IEE Proceedings-Communications*, 2005, 152(6): 1098–1102.

[6] Dai Yongmei, Chen N, Arnold Z Yan, *et al*.. Memory efficient decoder architectures for quasi-cyclic LDPC codes [J]. *IEEE Transactions on Circuits and Systems*, 2008, 55(9): 2898–2911.

[7] Huang Q, Kang J, Zhang L, *et al*.. Two reliability-based iterative majority-logic decoding algorithms for LDPC codes [J]. *IEEE Transactions on Communications*, 2009, 57(12): 3597–3606.

[8] Lee J K and Thorpe J. Memory-efficient decoding of LDPC codes [C]. IEEE International Symposium on Information Theory, Adelaide, Australia, 2005: 459–463.

[9] Daesun Oh and Parhi K K. Min-sum decoder architectures with reduced word length for LDPC codes [J]. *IEEE Transactions on Circuits and Systems*, 2010, 57(1): 105–115.

[10] 钟州, 李云洲, 孙引, 等. 基于 LDPC 码校验节点度的分类修正最小和算法[J]. 清华大学学报, 2009, 49(1): 45–49.
Zhong Z, Li Y, Sun Y, *et al*.. Check node degree-based modified min-sum decoding algorithm with classification for LDPC codes [J]. *Journal of Tsinghua University*, 2009, 49(1): 45–49.

[11] Zhong Z, Li Y, Chen X, *et al*.. Modified min-sum decoding algorithm for LDPC codes based on classified correction [C]. International Conference on Communications and Networking in China, Hangzhou, China, 2008: 932–936.

[12] 闫涛, 茹乐, 杜兴民. 一种基于折线逼近的对数似然比简化算法 [J]. 电子与信息学报, 2008, 30(8): 1832–1835.
Yan Tao, Ru Le, and Du Xing-min. A simplified log-likelihood ratio algorithm with broken line analysis [J]. *Journal of Electronics & Information Technology*, 2008, 30(8): 1832–1835.

[13] IEEE Std 802.16e: Air interface for fixed and mobile broadband wireless access systems [S]. IEEE LAN/MAN Standard Committee. New York: IEEE, 2004.

马　卓：　男，1981 年生，讲师，研究方向为无线通信中的信道估计、信道均衡和信道编码技术.

杜栓义：　男，1960 年生，教授，硕士生导师，研究方向为无线通信抗干扰与软件无线电.

王新梅：　男，1937 年生，教授，博士生导师，研究方向为通信、纠错码和密码.