

基于改进的整数量化 LDPC 码分层译码算法

劳玲玲, 李立欣, 朱梦, 张会生

(西北工业大学 电子信息学院, 陕西 西安 710129)

摘要: LDPC 码分层译码算法在进行整数量化操作时, 存储单元的限制会导致译码信息的溢出。本文系统分析了溢出错误的原因, 并提出了两种改进的分层译码方案, 错误部分消除方案 (partially eliminating errors scheme, PEES) 和不同比特量化方案 (different bit quantization scheme, DBQS)。两种改进方案分别从消除部分错误和避免溢出错误的角度来改进译码性能, 且硬件实现时只需增加一定数量的加法器和移位操作。通过对不同码长、不同量化比特的 LDPC 码进行仿真, 结果表明, 2 种方案均有效地抑制了溢出错误, 与基于全精度浮点数运算的修正算法相比, 在误码率为 1.0×10^{-4} 时, 分别仅有约 1 dB 和 1.8 dB 的性能损耗。

关键词: 低密度奇偶校验码, 分层译码, 整数量化, 溢出错误

中图分类号: TN911.22

文献标志码: A

文章编号: 1000-2758(2014)06-0974-06

LDPC (low-density parity-check) 码作为一种特别的纠错码, 相对 Turbo 码而言, 译码算法简单, 具有较高的吞吐量和良好的译码性能, 并且 LDPC 码的译码算法可实现完全并行的操作, 具备实现高速译码的潜力。LTE-Advanced 系统要求 1 Gbit/s 的传输速率, LDPC 码将会是比较理想的纠错编码选择。传统的 LDPC 码译码算法研究中基本采用高精度浮点数进行运算, 但在实际的硬件实现中高精度浮点数不容易实现。基于此, 很多学者提出基于整数量化的 LDPC 码译码算法, 文献 [1] 提出一种 MRBI-MLGD 译码算法, 对接收到的信息 y 进行 b 比特均匀量化以实现对信息的整数运算处理。文献 [2] 提出一种 IDRb-MLGD 的非二进制 LDPC 码译码算法, 对纠错性能和译码复杂度有效折中。文献 [3] 提出一种 MISRB-MS 译码算法, 算法引进了乘性因子, 虽然增加了一些乘法运算, 但实现了更好的性能。这些算法虽然一定程度地降低了硬件实现的复杂度, 但是 BP 算法需要更新全部的校验信息和变量信息, 这需要大量的运算, 且正确译码需要的迭代次数比较大, 这会使整体的译码速度降低。

基于此, LDPC 码的分层译码算法^[4]被提出, 与

BP 算法相比, 该算法具有更快的收敛速度、更低的硬件实现复杂度、更少的存储单元等优点。但是在对信息进行量化处理后, 该算法的译码性能大大降低, 为解决这些问题许多基于整数量化的分层译码算法被提出^[5-9]。同时许多学者开始致力于 LDPC 码分层译码器的设计^[10-12]。文献 [13] 提出了一种基于图的近似最优计划方案, 该方案允许多行运算, 并克服了分层译码算法中寄存器争用的问题。但分层译码算法在进行整数量化操作时存在溢出现象, 这会造成严重的译码错误, 对该现象进行分析并采取措施解决该问题是我们需要进行研究的。对 LDPC 码的分层译码算法进行研究与改进, 在保证较好性能的前提下设计复杂度低适于硬件实现的译码算法, 能够解决 4G 通信系统中实时性、大容量数据传输的瓶颈, 提高信息传输的可靠性。

1 LDPC 码分层译码算法

LDPC 码分层译码算法 (TDMP) 将校验矩阵 H 以行为单位划分为几个水平分层, 每一个分层包含

收稿日期: 2014-04-16

基金项目: 中央高校基本科研业务费专项资金 (3102014JCQ1052)、上海航天科技创新基金

(SAST201455) 与中国博士后科学基金 (2014M552490) 资助

作者简介: 劳玲玲 (1989—), 女, 西北工业大学硕士研究生, 主要从事信道编码研究。

所有的变量节点和该层的校验节点, 基于 BP 算法的置信传播思想, 每一个分层都视为一个子模块对其进行迭代更新。当一个分层的所有校验节点完成信息更新后, 将校验消息作为外部信息传递给变量节点进行一次更新, 并传递到下一层; 下一层利用上一层更新的变量节点的后验信息和校验节点的外部信息更新变量节点传递给校验节点的先验信息, 进行该层的迭代更新。当所有分层更新完成后整个译码过程完成一次迭代。与 BP 译码算法相比, 分层译码算法每一层都对变量信息进行更新, 因此具有更快的收敛速度; 同时分层译码算法具有硬件实现复杂度低、所需存储单元少等优点。

为了便于硬件实现, 降低运算复杂度, 本文在对变量节点处理时运用 BP 算法中的最小和译码算法 (min-sum algorithm, MSA) 的变量信息处理方法, 即 LDPC 码 TDMP-MSA 译码算法^[14-15], 同时采用整数量化方案, 将所有的信息从浮点数转化为整数进行运算, 更进一步地降低运算复杂度和减少存储单元数量。本文采用 q 比特均匀量化方案, q 的地位为符号位, 其他位为量化范围, 则量化范围为 $Z = 2^{q-1} - 1$, 对初始信息 y_n 进行量化所得量化信息 Y_n 为

$$Y_n^{(0)} = \begin{cases} Z & y_n \geq 3.2 \\ \text{round}\left(\frac{y_n}{\Delta}\right) & |y_n| < 3.2 \\ -Z & y_n \leq -3.2 \end{cases} \quad (1)$$

式中: $\Delta = 3.2/Z$, round 表示进行四舍五入取整运算, 这样操作无需进行大量的查表操作便可快速实现整数量化。

本文采用 LDPC 码的 TDMP-MSA 译码算法, 将校验矩阵 $H_{M \times N}$ 划分为 K 层, 在每一层对信息进行更新处理, 并及时将更新的变量消息传递到下一层, 从而使变量消息的收敛速度大大加快, 减少译码时间, 提高系统吞吐量。一次完整的迭代过程如下:

- 1) 初始化(校验节点 m , 变量节点 n)

$$\gamma_n^{(0,1)} = \gamma_n^{(0)} = Y_n \quad (2)$$

$$\lambda_n^{(0)} = 0 \quad (3)$$

- 2) 一次子迭代过程(1 为迭代次数, i 为层数)

- a) 校验节点 m 向变量节点 n 传递消息

$$\rho_{mn}^{(l,i)} = \gamma_n^{(l,i-1)} - \lambda_{mn}^{(l-1)} \quad (4)$$

$$\lambda_{mn}^l = \prod_{n' \in N(m)/n} \text{sign}(\rho_{mn'}) \cdot \min_{n' \in N(m)/n} |\rho_{mn'}| \quad (5)$$

- b) 变量节点 n 信息的更新

$$\gamma_n^{(l,i)} = \rho_{mn}^{(l,i)} + \lambda_{mn}^{(l)} \quad (6)$$

- 3) 重复过程 B, 至最后一个分层完成第 l 次迭代

- 4) 译码判决

$$x_n = \begin{cases} 1, & \gamma_n \leq 0 \\ 0, & \gamma_n > 0 \end{cases} \quad (7)$$

λ_{mn} 表示校验节点 m 传送到变量节点 n 的外部信息, ρ_{mn} 表示变量节点 n 传送到校验节点 m 的先验信息, γ_n 表示一次子迭代后得到变量节点 n 的后验信息。

基于浮点数运算的 LDPC 码分层译码算法无需对初始信息进行量化操作, 在校验节点处理单元运用最小和译码算法时, 近似运算实际上对 λ_{mn}^l 进行了放大, 现有 2 种改进算法^[16] 对 λ_{mn}^l 的值进行修正。一种方法为引入偏移量, 对 λ_{mn}^l 进行修正, 如式 (8) 所示:

$$\lambda_{mn}^l = \prod_{n' \in N(m)/n} \text{sign}(\rho_{mn'}) \cdot \max\left(\left(\min_{n' \in N(m)/n} |\rho_{mn'}| - \beta\right), 0\right) \quad (8)$$

β 为偏移因子。但是这种方法并不是最优的, 另一种方法为引入修正因子, 对 λ_{mn}^l 进行修正, 如 (9) 式所示:

$$\lambda_{mn}^l = \alpha_1 \cdot \prod_{n' \in N(m)/n} \text{sign}(\rho_{mn'}) \cdot \min_{n' \in N(m)/n} |\rho_{mn'}| \quad (9)$$

式中: α_1 为修正因子, 该修正因子的最优值为 0.8^[16-17], 限于篇幅, 本文不再深入研究偏移因子值的选取。目前普遍使用引入偏移因子的方法, 本文仿真时将用该方法作为参考方法。

2 译码错误的分析

LDPC 码分层译码算法当采用整数量化操作时, 相对浮点数运算速度大大提高, 但是有限位的存储单元限制了量化范围, 当信息值超过某一门限值时, 大于门限的部分将会溢出, 引起译码性能损失, 甚至导致译码失败, 本部分将深入分析溢出错误的产生原因。将校验矩阵 $H_{M \times N}$ 划分为 K 层, 每一层分别表示为 (m_1, m_2, \dots, m_K) , 校验节点 m 和变量节点 n 之间进行信息传递, 由 (4) 式和 (6) 式可以得到第 l 次迭代, 每层变量节点的后验消息如 (10) 式所示:

$$\begin{aligned} \gamma_l^{m_1} &= \gamma_l^{m_K} - \lambda_{l-1}^{m_1} + \lambda_l^{m_1} \\ \gamma_l^{m_2} &= \gamma_l^{m_1} - \lambda_{l-1}^{m_2} + \lambda_l^{m_2} \\ &\vdots \end{aligned}$$

$$\gamma_i^{mK} = \gamma_{i-1}^{mK} + \sum_{i=1}^K (\lambda_i^{m_i} - \lambda_{i-1}^{m_i}) \quad (10)$$

由于译码器存储单元的限制,当 $\lambda_i^{m_i}$ 的值超过门限值 Z 时便会出现溢出错误,此时 $\lambda_i^{m_i}$ 的值变为 $\hat{\gamma}_i^{m_i}$, 如(11)式所示:

$$\hat{\gamma}_i^{m_i} = \text{Thre}(\gamma_i^{m_i})_Z = \gamma_i^{m_i} + \varepsilon_i^{m_i} \quad (11)$$

Thre 代表门限函数,当 $\lambda_i^{m_i}$ 的值超过门限值 Z 时 $\lambda_i^{m_i}$ 的值为 Z , $\varepsilon_i^{m_i}$ 为溢出错误。根据(10)式的关系,可以得出

$$\hat{\gamma}_i^{mK} = \text{Thre}(\gamma_i^{mK})_Z = \gamma_i^{mK} + \sum_{i=1}^K \varepsilon_i^{m_i} = \gamma_i^{mK} + \varepsilon_i \quad (12)$$

这些错误会引入到下一代的迭代中,则下一代的先验信息为:

$$\hat{\rho}_{i+1}^{m_1} = \text{Thre}(\gamma_i^{mK})_Z - \lambda_i^{m_1} = \gamma_i^{mK} - \lambda_i^{m_1} + \varepsilon_i$$

$$\hat{\rho}_{i+1}^{m_2} = \text{Thre}(\gamma_{i+1}^{m_1})_Z - \lambda_i^{m_2} = \rho_{i+1}^{m_2} + \varepsilon_i + \varepsilon_{i+1}^{m_2}$$

⋮

$$\hat{\rho}_{i+1}^{mK} = \rho_{i+1}^{mK} + \varepsilon_i + \sum_{i=2}^K \varepsilon_{i+1}^i \quad (13)$$

由(13)式可以看出上一次译码产生的错误将会被带到这次迭代中,并且每次每层迭代又会产生新的错误,随着迭代次数的累积,错误也将累加,这将严重影响译码性能,导致译码失败。

3 PEES 方案和 DBQS 方案

针对上述问题,本文提出 2 种解决方案,PEES 方案和 DBQS 方案。

首先介绍方案 PEES,将(13)式的部分错误消除,对译码性能进行改进。 $\hat{\gamma}_i^{m_i}$ 的值是由于门限值限制的,我们无法对其改变,本文从 $\lambda_i^{m_i}$ 出发,对其进行修正,具体方法如下:

$$|\lambda_i^{m_1}|_{\text{mod}} = \text{Thre}(\gamma_i^{m_1}) - \rho_i^{m_1} = \lambda_i^{m_1} + \varepsilon_i^{m_1}$$

$$|\lambda_i^{m_2}|_{\text{mod}} = \lambda_i^{m_2} + \varepsilon_i^{m_1} + \varepsilon_i^{m_2}$$

⋮

$$|\lambda_i^{mK}|_{\text{mod}} = \lambda_i^{mK} + \varepsilon_i \quad (14)$$

将(14)式带入(13)式,则

$$|\rho_{i+1}^{m_1}|_{\text{mod}} = \text{Thre}(\gamma_i^{mK})_Z - |\lambda_i^{m_1}|_{\text{mod}} = \rho_{i+1}^{m_1} + \varepsilon_i - \varepsilon_i^{m_1}$$

$$|\rho_{i+1}^{mK}|_{\text{mod}} = \rho_{i+1}^{mK} + \sum_{i=2}^K \varepsilon_{i+1}^i \quad (15)$$

将(15)式与(13)式对比,可以看出先验信息的错误被部分消除了,特别是在下一代迭代的第一层

先验信息更新时,错误完全来源于上一次的迭代错误,且由于 $\varepsilon_i^{m_1}$ 与 $\lambda_i^{m_1}$ 负相关, ε_i 中的 $\varepsilon_i^{m_1}$ 对第一层的信息更新结果影响很大,通过上述方案对 $\lambda_i^{m_i}$ 的修正可以消除部分错误,有效抑制译码失败概率,提高译码性能。PEES 方案第 l 次迭代第 m_i 层译码的实现过程如图 1 所示,其中 Thre 为门限设置,PCUB 为校验节点处理单元。但 PEES 方案只是消除了影响较大的那部分错误,公式(15)还存在错误,即当 $|\gamma_i^{m_i}| \geq Z$ 时,由于门限的限制,使得 $|\gamma_i^{m_i}| = Z$,此时 $|\rho_{i+1}^{m_i}|_{\text{mod}}$ 的值被缩小了,为了避免译码复杂,本文根据 BP 算法的改进算法,对 $|\rho_{i+1}^{m_i}|_{\text{mod}}$ 进行修正得到 $|\rho_i^{m_i}|_{\text{Norm}}$

$$|\rho_i^{m_i}|_{\text{Norm}} = \begin{cases} \text{Thre}(\gamma_i^{m_i})_Z - |\lambda_{i-1}^{m_i}|_{\text{mod}}, & |\gamma_i^{m_i}| < Z \\ \text{Thre}(\gamma_i^{m_i})_Z - \alpha \cdot |\lambda_{i-1}^{m_i}|_{\text{mod}}, & |\gamma_i^{m_i}| = Z \end{cases} \quad (16)$$

α 为正数且小于 1 的修正因子,为了便于硬件实现, α 的值取 1/2,只需将寄存器向右循环一位即可,不必增加其他的存储单元和逻辑单元,修正后的方案称为 PEES-Modify 方案。

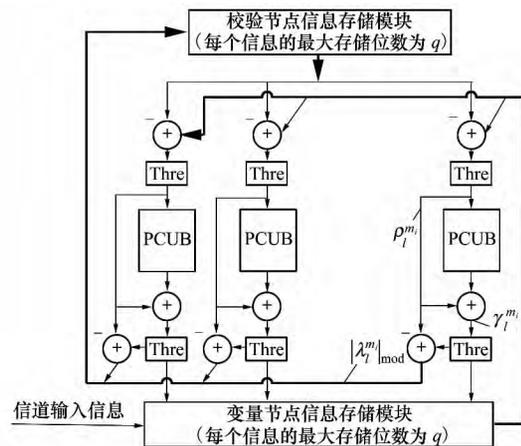


图 1 PEES 方案第 l 次迭代第 m_i 层译码实现框图

PEES 方案以消除错误为目的,DBQS 方案的思想为避免错误的出现。DBQS 方案的解决方法为设置不同比特量化范围,若存储单元为 q 位,则初始信息位设置为 $q-1$ 位, $\rho_i^{m_i}$ 的位数也设定为 $q-1$ 位,而 $\gamma_i^{m_i}$ 的位数为 q 。此时,即使 $\rho_i^{m_i}$ 和 $\lambda_i^{m_i}$ 同时都达到门限值, $\gamma_i^{m_i}$ 的值也不会超过门限值,这样避免了 $\gamma_i^{m_i}$ 值溢出错误的出现。但是当 $\rho_i^{m_i}$ 和 $\lambda_i^{m_i}$ 超出门限值时会引入一定的误差,且分层译码算法的最小和译码算法对 $\lambda_i^{m_i}$ 的值进行了放大处理,为了进一步减小性能损耗,本文基于浮点数运算算法的改进思想对 $\lambda_i^{m_i}$

进行修正, 令

$$\lambda_{l(mn)}^{m_i} = \alpha_2 \cdot \prod_{n' \in N(m)/n} \text{sign}(\rho_{l(mn')}^{m_i}) \cdot \min_{n' \in N(m)/n} |\rho_{l(mn')}^{m_i}| \quad (17)$$

式中: α_2 为修正因子, 为了便于硬件实现, α_2 的值设定为 1/2。虽然量化区间的缩小会导致一定的性能损耗, 但是通过不同量化区间的设定避免了 $\gamma_l^{m_i}$ 值溢出错误的出现, 且只须增加移位操作, 译码复杂度和硬件实现复杂度低。DBQS 方案第 i 次迭代第 m_i 层译码的实现框图如图 2 所示。

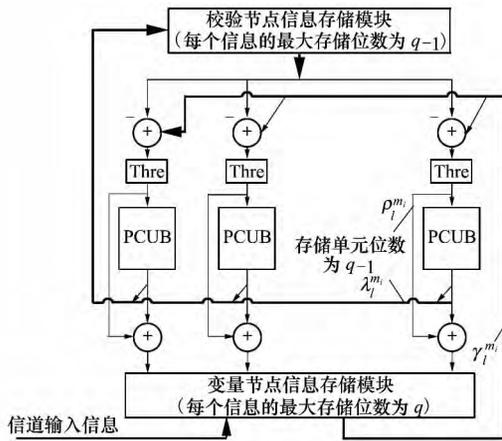


图 2 方案二第 l 次迭代第 m_i 层译码实现框图

DBQS 方案并未增加额外的加法器, 只是校验节点和变量节点的移位寄存器存储单元位数不同。以码率为 1/2 的 LDPC 码($n, p, 2p$) 为例, 各种方案每一次完整迭代的计算量如表一所示, 其中 $(n/2) \cdot (2p - 2 + \lceil \log_2^{2p} \rceil)$ 是校验节点最小和运算处理所需的加法运算次数。可以看出, 除校验节点处理单元之外, DBQS 方案所需加法器的数量是 PEES 方案的 2/3。

表 1 各种方案计算复杂度比较

方案	加法运算	移位操作
PEES 方案	$\frac{n}{2}(2p - 2 + \lceil \log_2^{2p} \rceil) + 3np$	
PEES-Modify 方案	$\frac{n}{2}(2p - 2 + \lceil \log_2^{2p} \rceil) + 3npmp (0 \leq m \leq n)$	
DBQS 方案	$\frac{n}{2}(2p - 2 + \lceil \log_2^{2p} \rceil) + 2np$	np

4 性能仿真

本文采用由随机构造法构造的 LDPC 码, 码型分别为(256, 512), (512, 1024), 即码率均为 1/2 的

LDPC 码, 对 1 000 帧数据进行仿真, 仿真信道为高斯白噪声信道, 对不同码型、不同译码算法、不同比特量化、不同迭代次数下的 LDPC 码译码性能进行仿真和分析, 以证明本文提出的算法对 LDPC 性能上的改进。其中基于浮点数运算的分层译码算法表示为 Float-Layered, 基于整数量化运算的分层译码算法表示为 Layered-Quantization。

图 3 为 LDPC 码(256, 512) $q=6$ 比特量化, 最大迭代次数为 5 的条件下, 不同分层译码算法仿真图。由图 3~图 6 可以看出: 当只采用整数量化方案时, 译码失败, 误码率曲线接近 1, 这是由溢出错误的出现和译码时错误的叠加导致; 为解决这种现象, 本文提出了 2 种方案, 其中 PEES 方案有效地消除了主要错误, 而 PEES-Modify 方案进一步对性能进行改进, 当误码率为 10^{-4} 时, 与全精度浮点数运算的分层译码算法相比, 分别具有约 1.3 dB 和 1 dB 的损耗; 而 DBQS 方案也有效地避免了溢出错误的出现, 当误码率为 10^{-4} 时, 与 PEES 方案相比, 具有约 1 dB 的损耗。

图 4 为 LDPC 码(256, 512) $q=8$ 比特量化, 最大迭代次数为 5 的条件下, 不同分层译码算法仿真图。与图 3 相比, 可以看出: 当量化比特变大时, 译码性能更好, 这是因为量化比特变大, 量化区间变大, 溢出错误的出现概率变小; 本文提出的 2 种方案均有效地对整数量化分层译码算法的进行了性能的改进。PEES 方案和 PEES-Modify 方案在误码率为 10^{-4} 时, 与全精度浮点数运算的分层译码算法相比, 分别具有约 1 dB 和 0.5 dB 的损耗; 而 DBQS 方案与 PEES 方案相比, 具有约 0.5 dB 的损耗。虽然 DBQS 方案与 PEES 方案相比有一定的性能损耗, 但 DBQS 方案的实现方法简单, 复杂度较低, 通过 2 种方案计算量的比较, 如表一所示, 可以看出 PEES-Modify 方案与基本算法相比需要额外的移位操作。

图 5 为 LDPC 码(512, 1024) $q=6$ 比特量化, 最大迭代次数为 5 的条件下, 不同分层译码算法仿真图。与图 3 相比, 可以看出: 码长增加时, 译码性能会提高, 无论哪种码型, 本文提出的 2 种方法均可以有效消除部分溢出错误或避免错误, 从而使基于整数量化的分层译码算法的性能得到改进。

图 6 为 LDPC 码(256, 512) $q=8$ 比特量化, 最大迭代次数为 10 的条件下, 不同分层译码算法仿真图。与图 4 相比, 当迭代次数增加时, 基于浮点数运算的分层译码算法的性能, 在误码率为 10^{-4} 时有约

0.2 dB 的提高,但由于门限溢出现象,溢出错误会叠加,基于整数量化运算算法的译码性能均会有所下降。本文中 choice 迭代次数为 5。

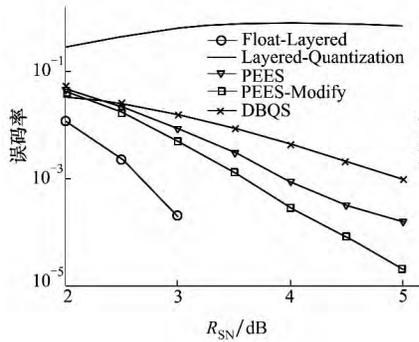


图 3 LDPC(256, 512) $q=6$, $iter=5$ 不同译码性能比较

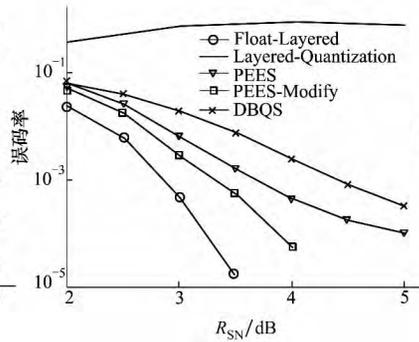


图 4 LDPC(256, 512), $q=8$, $iter=5$ 不同译码性能比较

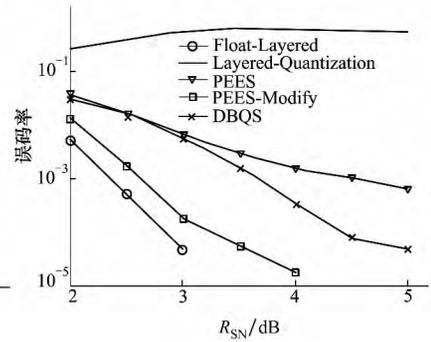


图 5 LDPC(512, 1024), $q=8$, $iter=5$ 不同译码性能比较

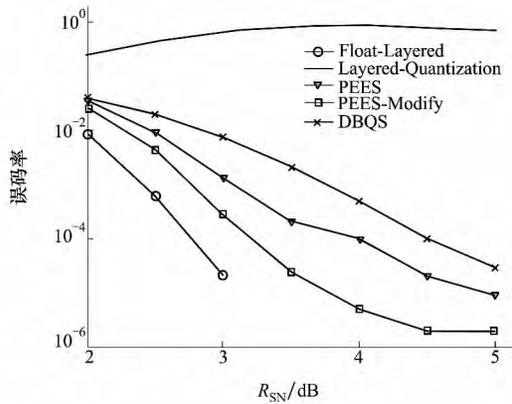


图 6 LDPC(256, 512) $q=8$, $iter=10$ 不同译码性能比较

5 结 论

本文通过对基于整数量化运算的 LDPC 码分层译码算法的数学公式中溢出错误的分析和计算,分析了产生溢出错误的原因,针对该问题,提出了 2 种解决方案,消除部分错误方案 (PEES) 和不同比特量化方案 (DBQS)。PEES 方案通过对溢出错误中的主要错误部分进行消除有效地避免了译码的失败,并提出了 PEES-Modify 方案,对译码性能进一步改

善。仿真结果表明,当迭代次数为 5,量化比特为 8 时,与基于浮点数运算的 LDPC 码分层译码算法相比,在误码率为 10^{-4} 时,PEES 方案仅有约 1 dB 的性能损耗,且硬件实现时 PEES 方案只需对传统基于整数量化的分层译码算法增加一定数量的加法器;PEES-Modify 方案在 PEES 方案的基础上增加一定数量移位操作,但改进方案的性能优于基本方案,在误码率为 10^{-4} ,PEES-Modify 方案相比 PEES 方案有 0.5 dB 的性能提高。DBQS 方案从避免溢出错误的角度出发,对初始信息和先验信息采用少于存储单元一位的量化比特,而后验信息的量化范围为所规定的量化比特,仿真结果表明,该方案与 PEES 方案相比在误码率为 10^{-4} 时,仅有约 0.8 dB 的性能损耗,与基于浮点数运算的算法相比,性能损失约 1.8 dB,但该方案只需对传统基于整数量化的分层译码算法增加一定数量的移位操作,硬件实现复杂度低于 PEES 方案。总之,本文提出的 2 种译码方案均有效避免了基于整数量化运算的 LDPC 码分层译码算法中溢出错误的出现,仿真结果证明了 2 种方案的可行性,PEES 方案的译码性能优于 DBQS 方案,但 DBQS 方案的译码复杂度低于 PEES 方案,在实际应用中可根据译码性能和译码复杂度的不同要求选择相应的方案。

参考文献:

- [1] Chen Haiqiang, Li Jinfeng, Qin Tuanfa. The Modified Reliability-Based Iterative Majority-Logic Decoding Algorithm for Structured LDPC Codes [C]//International Conference on Electronic & Mechanical Engineering and Information Technology, 2011: 2873-2875
- [2] Lu Yichao, Qiu Nanfan, Chen Zhixiang. An Efficient Majority-Logic Based Message-Passing Algorithm for Non-binary LDPC Decoding [C]//IEEE Asia Pacific Conference on Circuits and Systems, 2012: 479-482
- [3] Zhang Zuotao, Fang Yibo, Liu Guanghui. Two Efficient Algorithms Based on Majority-Logic and Min-Sum Algorithms for LDPC Codes [C]//The 8th International Conference on Wireless Communications, Networking and Mobile Computing, 2012: 1-4
- [4] Mansour M M. A Turbo-Decoding Message-Passing Algorithm for Sparse Parity-Check Matrix Code [J]. IEEE Trans on Signal Process, 2006, 54(11): 404-412
- [5] 张嵩, 马林华, 唐红 等. 基于整数运算的 LDPC 码改进分层译码算法 [J]. 系统工程与电子技术, 2013, 35(3): 638-642
Zhang Song, Ma Linhua, Tang Hong. Improved Layered Decoding Algorithm for LDPC Codes Based on Integer Operation [J]. Systems Engineering and Electronics, 2013, 35(3): 638-642 (in Chinese)
- [6] 杨威, 张为. 一种基于分层译码和 Min-max 的多进制 LDPC 码译码算法 [J]. 电子与信息学报, 2013, 35(7): 1677-1681
Yang Wei, Zhang Wei. A Decoding Algorithm Based on Layered Decoding and Min-max for Nonbinary LDPC Codes [J]. Journal of Electronics & Information Technology, 2013, 35(7): 1677-1681 (in Chinese)
- [7] JaWone A Kennedy, Daniel L Noneaker. A Technique to Improve the Performance of Fixed-Point TDMP Decoding of QC-LDPC Codes in the Presence of SNR Estimation Error [C]//IEEE Conference on Military Communications, 2011: 649-654
- [8] Han Guojun, Liu Xingcheng. An Efficient Dynamic Schedule for Layered Belief-Propagation Decoding of LDPC Codes [J]. IEEE Communication Letters, 2009, 13(12): 950-952
- [9] Vila Casado, Andres I, Miguel Griot. Improving LDPC Decoders via Informed Dynamic Scheduling [C]//IEEE Information Theory Workshop, 2007: 208-213
- [10] 张宇阳, 胡剑浩, 李峰. 符合 DMB-T 标准的 TDMP-LDPC 译码器设计 [J]. 中国集成电路设计, 2009, 115: 26-31
Zhang Yuyang, Hu Jianhao, Li Feng. A TDMP-LDPC Decoder Designed for DMB-T Standard [J]. China Integrated Circuit, 2009, 115: 26-31 (in Chinese)
- [11] 鲍丹, 向波, 申睿, 等. 基于 TDMP 优化算法的 QC-LDPC 译码器 VLSI 实现 [J]. 计算机研究与发展, 2009, 46(2): 338-344
Bao Dan, Xiang Bo, Shen Rui, et al. VLSI Implementation of QC-LDPC Decoder Using Optimized TDMP Algorithm [J]. Journal of Computer Research and Development, 2009, 46(2): 338-344 (in Chinese)
- [12] 黄双渠. 多模式 LDPC 译码器算法研究和 VLSI 实现 [D]. 上海: 复旦大学, 2010
Huang Shuangqu. Research and VLSI Implementation of LDPC Decoder Algorithm in Multi-Patterns [D]. Shanghai, Fudan University, 2010 (in Chinese)
- [13] Ismail M, Coon J, Ahmed I. High Throughput Layered Decoding of LDPC Codes [C]//IEEE International Symposium on Personal, Indoor and Mobile Radio Com, 2009: 1727-1731
- [14] Mohammad M Mansour. A Turbo-Decoding Message-Passing Algorithm for Sparse Parity-Check Matrix Codes [J]. IEEE Trans on Signal Processing, 2006, 54(11): 4376-4392
- [15] Mohammad M Mansour, Shanbhag N R. A 640Mb/s 2048 Bit Programmable LDPC Decoder Chip [J]. IEEE Journal on Solid-State Circuits, 2006, 41(3): 684-698
- [16] 袁东风, 张海刚. LDPC 码理论与应用 [M]. 北京: 人民邮电出版社, 2008: 84-89
Yuan Dongfeng, Zhang Haigang. Theory and Applications of LDPC Codes [M]. Beijing, Posts & Telecom Press, 2008: 84-89 (in Chinese)
- [17] 蔡丽萍, 马晓伟, 李俊红. 基于电力线通信的 LDPC 码分层译码算法研究 [J]. 微电子学与计算机, 2013, 30(5): 125-128
Cai Liping, Ma Xiaowei, Li Junhong. A Study of LDPC Codes Layered Decoding Algorithm on Power Line Communications [J]. Microelectronics and Computer, 2013, 30(5): 125-128 (in Chinese)

Improved Layered Decoding Algorithms Based on Integer Quantization for LDPC Codes

Lao Lingling , Li Lixin , Zhu Meng , Zhang Huisheng

(Department of Electronics Engineering , Northwestern Polytechnical University , Xi'an , 710129 , China)

Abstract: When layered decoding algorithm for LDPC (Low-Density Parity-Check) codes is performing integer quantization operation , restriction of storage units will cause the problem of decoding information overflowing. This paper systematically analyzes the reason why overflow errors occur , and proposes two improved layered decoding schemes: Partially Eliminating Errors Scheme (PEES) and Different Bit Quantization Scheme (DBQS) . Two improved schemes improve decoding performance by partially eliminating some errors and avoiding overflow errors respectively , and hardware implementation requires only some extra summators and shift operations. In this paper , we simulate the performance of the LDPC codes in different code lengths and different quantization bits , the simulation results and their analysis show preliminarily that: both programs both effectively suppress overflow errors; compared with the modified algorithm based on full-precision floating-point operations , the two proposed schemes can achieve a performance loss of about only 1 dB and 1.8 dB respectively around BER = 10⁻⁴.

Key words: belief-propagation algorithms , bit error rate; integer quantization , layered decoding , LDPC(low-density parity-check) , overflow errors



《机械科学与技术》2014 年第 33 卷全年英摘关键词平均每篇使用 21 世纪 Ei 主题词 1.39 个

《机械科学与技术》2014 年第 33 卷全年共有 389 篇论文 ,共使用 541 个 21 世纪 Ei 主题词 ,平均每篇 1.39 个。

全年使用 21 世纪 Ei 主题情况如下表:

期或季度	1 季度	2 季度	3 季度	第 10 期	第 11 期	第 12 期	4 季度	全年
篇数	91	96	102	32	32	36	100	389
共使用 21 世纪 Ei 主题词数目	99	128	133	54	61	66	181	541
平均每篇使用的 21 世纪主题词数目	1.09	1.33	1.30	1.69	1.91	1.83	1.81	1.39

胡沛泉
2014 年 12 月