



RS-LDPC 概率联合译码算法

孙怡宁*, 黄秋, 胡剑浩*, 康凯

电子科技大学通信抗干扰技术国家级重点实验室, 成都 611731

* 通信作者. E-mail: ynsun3842@163.com, jhhu@uestc.edu.cn

收稿日期: 2020-10-13; 修回日期: 2020-11-23; 接受日期: 2021-01-06; 网络出版日期: 2022-05-13

国家重点研发计划 (批准号: 2018YFB1801500) 资助项目

摘要 现有的译码方案中, 由于复杂度问题, LDPC 码多使用部分并行译码架构, 难以满足高速、超高速应用; RS 码多采用硬判决译码算法. 因此, 对于 RS-LDPC 级联码译码在复杂度和性能上有进一步优化的空间. 本文基于概率计算的思想提出了 RS-LDPC 概率联合迭代译码算法. 该算法继承了概率 LDPC 译码器的低复杂度优势和概率 RS 译码算法可采用硬译码器逼近软译码性能的特点, 同时通过联合迭代译码架构改善了译码性能. 为此, 我们设计了一种基于 LDPC 概率值的测试向量产生方法和一种新颖的应用于概率 LDPC 译码器的变量节点结构的附加外信息生成机制. 仿真结果表明, 在误比特率为 10^{-6} 处, 我们提出的算法与使用浮点 BP 和 BM 硬判决译码的 RS-LDPC 级联译码方案相比, 可以获得 0.3 dB 的增益; 其实现复杂度为一个概率 LDPC 译码器和多个 RS 硬译码器的复杂度; 译码时延由概率 LDPC 译码器时延决定. 因此, 本文提出的联合译码算法具有很好的应用价值.

关键词 RS 码, LDPC 码, 概率计算, 联合迭代译码

1 引言

Forney 于 1966 年提出的级联编码^[1] 可以解决译码器实现复杂度和硬件开销随码长增加而增大的问题; 使用非二进制码 (应用中常用 Reed-Solomon (RS) 码) 作为外码, 二进制码作为内码的单级级联码已广泛应用于通信和数据存储系统中, 例如 CCSDS, CMMB, DVB-S2 等标准. 然而, 现有的 RS-LDPC 级联译码方案由于译码复杂度问题在实际应用中仍然受限.

对于 RS 码的译码方案, 硬判决算法因具有低复杂度的优势而得到广泛应用, 但其译码性能远差于软判决算法. Chase^[2] 算法是一种常见的软判决译码算法, 但其实现复杂度会随码长增加呈指数增长, 无法得到实际应用. 为了解决该问题, 基于概率计算的方法, Leroux 等^[3] 提出的比特级概率 Chase 算法 (bit-wise stochastic chase algorithm, BSCA) 和 Mani 等^[4] 提出的符号级概率 Chase 算

引用格式: 孙怡宁, 黄秋, 胡剑浩, 等. RS-LDPC 概率联合译码算法. 中国科学: 信息科学, 2022, 52: 922–933, doi: 10.1360/SSI-2020-0325

Sun Y N, Huang Q, Hu J H, et al. RS-LDPC stochastic joint decoding algorithm (in Chinese). Sci Sin Inform, 2022, 52: 922–933, doi: 10.1360/SSI-2020-0325

法 (symbol-level stochastic chase algorithm, SSCA) 大大降低了 Chase 软判决算法的实现复杂度, 但 BSCA 和 SSCA 的实现复杂度和译码时延依然远高于传统的 RS 硬判决译码算法。

对于 LDPC 译码算法, 置信度传播 (belief propagation, BP) 译码算法是目前常用的软译码算法。但 BP 算法涉及大量非线性计算, 硬件实现复杂度极高。为了简化 BP 算法的运算量, Fossorier 等^[5]提出了一种最小和算法 (min-sum algorithm, MSA), 在性能损失可接受范围内简化了校验节点的计算, 但是其实现复杂度依然难于支持全并行工作模式。随后, 学者们使用概率计算^[6]的方法来提高硬件效率。在基于概率计算的 LDPC 译码器设计方案中, 将变量节点 (variable node, VN) 和校验节点 (check node, CN) 之间传递的软信息转换为伯努利 (Bernoulli) 0, 1 比特流, 使数据宽度由多比特减小为 1 比特; 并且校验节点可以使用仅占用很少资源的异或门实现^[7,8], 这极大地减少了布线资源。研究表明^[9,10], 概率 LDPC 译码器在获得逼近 MSA 性能的条件下, 其硬件效率较 MSA 提高了 1 个数量级左右。因此, 概率 LDPC 译码器具有重要的研究和应用价值。

现有的大部分 RS-LDPC 译码方案是级联译码, 即内外码的译码模块之间没有信息的双向交互。Shi 等^[11]采用基于 BP 算法和 Chase 算法的联合译码, 在 LDPC 译码器与 RS 译码器之间增加了反馈, 相比使用 BP 算法的单独 LDPC 码获得了 0.5 dB 的增益。然而根据上述介绍, 该方法因其实现复杂度极高而无法在实际系统中得到应用。在最新的研究中, Qiu 等^[12]针对特定的 RS-SC-LDPC 码使用了基于 SWD 算法和 BM 算法的联合译码, 但因其交换的是软信息仍具有较高的硬件实现复杂度。

为了改善 RS-LDPC 级联码的译码性能, 同时降低其硬件实现复杂度, 我们利用概率算法低硬件复杂度的优势设计了基于概率 LDPC 译码器和 RS 硬译码器结构的联合译码方案, 包括测试向量生成方法、AEI 生成方法、具有 AEI 的变量节点架构和联合的译码提前终止机制的关键设计。在提出的方案中, 根据概率 LDPC 译码器中变量节点的概率跟踪值生成 RS 硬译码器的译码测试向量, 使得 RS 硬译码器获得逼近软译码算法的性能。同时, 将 RS 硬译码器结果作为附加外信息 (additional extrinsic information, AEI) 传递给概率 LDPC 译码器的变量节点, 以改善 LDPC 的信息更新性能。另外, 我们在联合译码架构中设置了两处提前终止机制来进一步降低译码时延。与现有的联合迭代算法相比, 我们使用了具有硬件可行性的概率译码结构。另外本质上的不同是, 在 RS-LDPC 的外迭代中, 我们的设计使 LDPC 译码模块合理地利用了处理后的 RS 正确译码信息, 而不是简单地将 RS 译码失败的码字送回 LDPC 译码再次进行译码尝试。仿真结果表明, 我们提出的算法与使用浮点 BP 和 BM 硬判决译码 (FPBP-BM) 的 RS-LDPC 级联译码方案相比可以获得 0.3 dB 的增益, 复杂度为一个概率 LDPC 译码器和多个 RS 码 BM 硬译码器的复杂度之和。时延方面, 联合译码器架构的译码时延由概率 LDPC 译码器时延决定, 并没有增加 RS 的译码时延。因此, 本文提出的联合译码算法具有很好的应用价值。

本文结构安排如下: 第 2 节简要阐述基于概率计算的 RS 译码方案和 LDPC 译码方案的设计原理, 并说明概率 RS-LDPC 联合译码的可行性; 第 3 节详细介绍提出的联合迭代译码方案的整体框架和设计细节; 第 4 节展示和分析仿真结果; 最后一节总结全文。

2 RS 码与 LDPC 码的概率译码算法

本节首先阐述现有的概率 RS 译码器和 LDPC 译码器, 并分析其工作原理和特性, 以阐述概率 RS-LDPC 联合译码的设计来源。

对于概率 RS 译码算法, 文献 [3,4] 中描述的 BSCA 和 SSCA 算法的核心思想都是将接收符号的概率与随机数比较产生测试向量, 通过大量测试向量的译码实验获得正确译码输出。仅使用概率生成模块和随机数发生器代替了 Chase 软判决算法, 通过翻转最不可靠比特产生测试向量的过程, 大大降

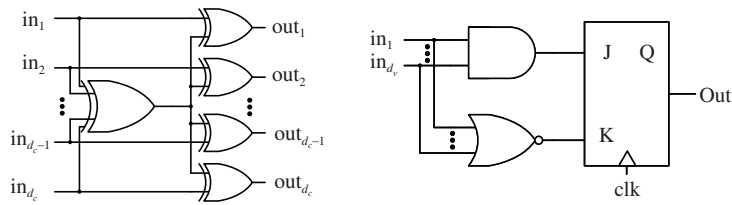


图 1 (a) 校验节点硬件结构图^[10]; (b) 变量节点硬件结构图
 Figure 1 (a) Structure of check nodes; (b) structure of variable nodes

低了 Chase 算法的运算复杂度和硬件实现复杂度. 概率 Chase 译码算法不仅简化了传统 Chase 软译码算法的实现, 而且依靠接收信号的置信度产生多个测试向量, 为正确译码提供更多的机会, 从而用较少的测试向量获得很好的译码性能, 以较低的运算复杂度和译码延迟获得了比传统 Chase 算法更好的译码性能. 因此, 本文基于 BSCA 的思想, 设计了一种测试向量生成方法, 为 BM 硬判决译码器提供合适的测试向量以获得软译码算法的性能.

基于概率计算实现的 LDPC 译码算法本质和 MSA 是一样的, 都是通过变量节点和校验节点之间的信息更新和传递来纠正变量节点的取值概率. 与 MSA 不同的是, 变量节点和校验节点之间传递的是能够表征概率值的比特序列而不是具体的概率值. 这使计算更加简单并且 Tanner 图中每条边只需传递一个比特值而不是用多个比特表示特定精度的概率值, 从而使硬件复杂度大大降低, 并且易于实现全并行结构. 变量节点和校验节点的结构都可以基于度 3 的结构组合而成, 其硬件结构图如图 1 所示, 可以看出概率 LDPC 译码器 VN 和 CN 的电路结构非常简单, 可以采用基本逻辑电路实现, 从而支持全并行工作模式. 因此, 概率 LDPC 译码器具有非常高的硬件效率. 此外, 概率 LDPC 译码器有一个非常重要的特点, 它是逐概率比特进行迭代更新的, 即一个时钟就可以完成一次迭代更新, 而 BP 算法需要多个时钟才可以完成一次迭代更新. 研究表明^[9, 13~18] 大多数概率 LDPC 译码器在最多译码周期为 400 DCs (decoding cycles), 平均译码周期为 30 DCs 的条件下, 就可以完成译码并获得接近 BP 算法的性能. 因此, 概率 LDPC 译码器具有非常高的吞吐率.

概率 LDPC 译码器存在的一个重要问题是变量节点的更新会产生锁死状态, 一旦变量节点输出的比特错误, 后续状态会持续输出错误比特, 导致译码性能变差, 特别是 Tanner 图中存在环的情况. 学者们提出了使用重随机模块^[7, 13~18] 来解决该问题, 当变量节点进入锁死状态时, 这时从重随机模块产生随机比特传递给校验节点. 本文采用 TFM^[14] 和 Saturated Up/down Counter^[17~19] (以下简称 counter) 两种架构进行仿真实验. 另外, 虽然概率 LDPC 译码算法硬件复杂度非常低, 但是相比 BP 算法仍有一定程度的性能损失, 例如对于本文采用的 (2048, 1723) LDPC 码, 在 BER 为 10^{-6} 处, 相比浮点 BP 仍有 0.15~0.35 dB 的性能损失.

根据对概率 RS 译码算法和概率 LDPC 译码算法的分析, 我们发现 RS 码的概率译码方案中输入硬判决译码器的测试向量可以由接收信号的 LLR 转换而成, 并且我们通过研究实验发现仅需要有限次的尝试即可得到正确码字, 从而能够以较低的复杂度和较快的收敛速度获得逼近软译码的性能. 同时, 概率 LDPC 译码器根据译码信息的跟踪概率生成判决比特输出, 这意味着该输出码字可以作为概率 RS 译码器的测试译码向量. 因此基于概率计算的 LDPC 和 RS 联合译码是可行的. 同时 RS 译码器的译码结果可以作为 AEI, 帮助 LDPC 译码器的变量节点从错误状态快速更新到正确状态, 从而改善概率 LDPC 译码器的译码性能和收敛速度. 因此, 联合迭代译码过程是一个平行的双迭代译码过程, 概率 LDPC 为内迭代译码过程, LDPC-RS 为外迭代译码过程. 但是联合译码方案的设计仍面临以下挑战:

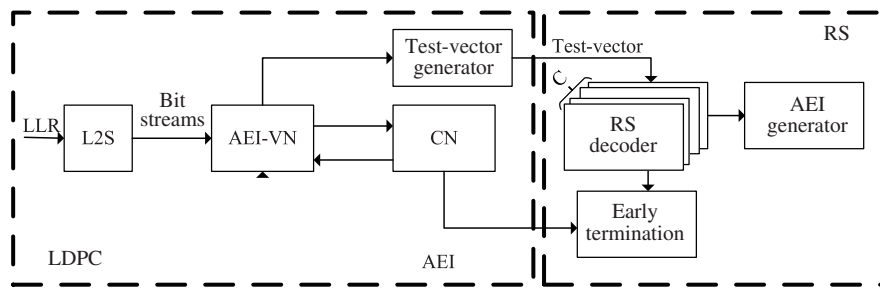


图 2 联合迭代译码器架构

Figure 2 Scheme of the joint iterative decoder

- (1) 如何通过概率 LDPC 译码器产生 RS 译码器的测试向量;
- (2) RS 译码器如何产生附加外信息;
- (3) 概率 LDPC 译码器中可处理附加外信息的变量节点的结构设计.

接下来我们将详细介绍解决上述问题的实现方法.

3 基于概率计算的联合迭代译码算法

在传统的 RS-LDPC 串行级联编译码系统中, RS 码与 LDPC 码的编译码器之间只是简单的级联, 采用流水的工作模式; 没有充分利用编码冗余信息和译码反馈信息, 无法获得联合译码增益. 因此在概率计算的思想, 我们提出了基于概率计算的 RS-LDPC 联合迭代译码算法, 其基本架构如图 2 所示. 其中, L2S 模块将 LLR 表征的信道信息转换为比特流送到 AEI-VN; AEI-VN 和 CN 构成 LDPC 码的迭代译码过程, AEI-VN 是我们设计的改进的 VN 结构, 增加了处理 AEI 的功能, CN 采用图 1(a) 的概率 LDPC 译码算法常用的简化结构; 在 LDPC 译码过程中, Test-vector Generator 将 AEI-VN 中的概率跟踪结果转换为 RS 码 BM 硬判决译码器的测试向量, BM 硬译码器根据测试向量进行译码尝试; AEI Generator 根据硬译码器输出形成 AEI 传递给 AEI-VN 作为反馈的附加外部信息以改善和加速其更新性能. 这样形成了内外并行的双迭代译码过程. 此外, 我们还利用 CN 的校验功能和 RS 硬译码器的译码结果形成译码提前终止机制, 使得译码过程可以在最大译码迭代周期前终止. 接下来将详细描述联合迭代译码实现的关键技术.

3.1 RS 测试向量生成模块

概率 LDPC 译码器在信息传递与迭代过程中具有译码周期 (DC), 即 VN 与 CN 之间更新和传递一次比特流信息记为 1 个 DC; 在硬件实现上通常 DC 与时钟周期一致. 如前所述, 由于概率 LDPC 译码器将多比特位宽降低为 1 比特位宽, 因此 1 个时钟即可完成一次信息的更新迭代, 而 BP 算法由于是多位宽运算, 一次信息的更新迭代需要多个时钟. 由于初始概率是等概率赋予的, 所以概率 LDPC 译码器在译码的初始阶段跟踪的概率是不能表征真实概率的, 在一定 DCs 后才会收敛; 并且如果比特发生翻转, 概率跟踪需要一个重新稳定的收敛过程. 因此, 如果 LDPC 译码结果每迭代一次都送入 RS 译码器进行译码, 在概率译码器的结构下会存在译码收敛过慢和错误扩散的问题, 从而影响译码性能. 因此我们希望 RS 译码器的第一个测试向量在 LDPC 译码器收敛到一定程度后产生, 此时需要的 DC 次数记为 DC_0 , 即首次内迭代译码周期; 在 LDPC 译码器处理 RS 译码器反馈的 AEI 信息后, 错误比特的概率需要经过一定的 DCs 才能收敛到新的正确状态, 这一过程所需要的 DC 次数记为 DC_t , 即内

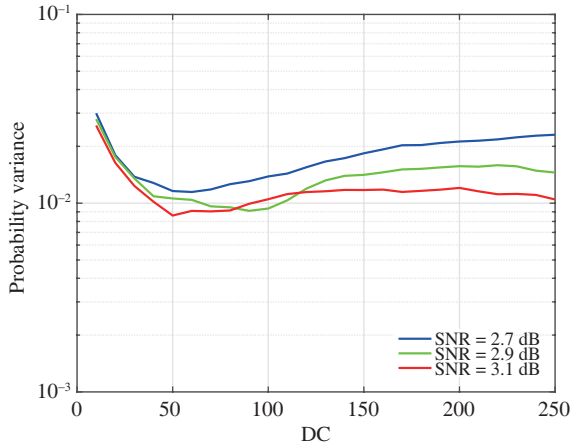


图 3 LDPC 概率译码器中跟踪概率的方差
Figure 3 Variance of tracking probability in LDPC stochastic decoder

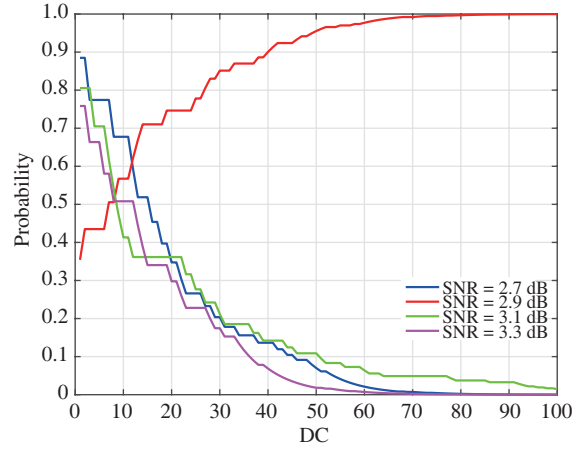


图 4 AEI 处理后 LDPC 概率译码器中概率的纠正过程
Figure 4 The correction process of probability in LDPC stochastic decoder after processing AEI

迭代译码周期. 寻找最佳 DC_0 和 DC_t 的方法如下.

假设二元 LDPC 码长为 N , 我们通过计算 VN 跟踪的概率方差寻找最佳 DC_0 . 令 P_c^i 表示第 i 个 VN 正确判决的概率, 即

$$P_c^i = \begin{cases} 1, & \text{if correct code bit is 1,} \\ 0, & \text{if correct code bit is 0,} \end{cases} \quad i = 0, 1, \dots, N - 1, \quad (1)$$

P_t^i 表示第 i 个 VN 节点跟踪的概率. 则 VN 跟踪概率的方差为

$$\sigma_P^2 = \frac{\sum_{i=0}^{N-1} |P_t^i - P_c^i|^2}{N}. \quad (2)$$

当方差趋于平稳, 说明 LDPC 概率译码器开始收敛, 此时可以向 RS 译码模块输出测试向量, 因此 DC_0 的取值为在跟踪概率方差平稳时对应的 DCs.

对于 DC_t , 即为统计跟踪概率越过 0.5 分界线需要的 DCs, 因为跟踪概率越过 0.5 分界线表征该比特进行了翻转, 状态得到了修正. 在传统 LDPC 概率译码器中, 错误比特的概率收敛到正确状态可能需要经过较多的 DCs, 这不是我们希望出现的. 因此在 3.3 小节我们提出了一种更有效的概率跟踪机制, 以减小 DC_t 来加快译码收敛速度.

基于上述理论依据, 结合仿真实验得到了使译码性能比较理想的 DC_0 和 DC_t 的取值范围, 如图 3 和 4 所示. 在图 3 中可以看出概率方差在 50 DCs 之后开始趋于平稳, 概率 LDPC 译码器开始处于稳定状态, 并且残留误码经过多次迭代无法消除, 因此推测 DC_0 可以从 50 DCs 开始取值; 在图 4 中, 经过 AEI 处理后, 大约 10~15 DCs 后概率值会越过 0.5, 说明比特信息被纠正, 在之后的迭代中概率值逐渐趋于稳定. 为了产生更多的测试向量为 RS 译码器提供更多的译码机会, 我们选择 $DC_t = 10$ DCs. 综上所述, 在本文的实验中, 概率 LDPC 译码器首先经过 DC_0 个译码周期后向测试向量产生器输出第一组比特序列, 此后每 10 个 DCs 向测试向量产生器输出一组比特序列.

3.2 附加外信息的生成

为了充分利用 RS 码的纠错能力、加快译码收敛速度,我们在 LDPC 译码器和 RS 译码器之间加入了附加外信息 (AEI) 产生器. 当 RS 译码成功并且通过附加的校验过程 (本文采用 CRC 校验) 时,我们认为 RS 译码器输出了正确的译码码字,此时将这段正确译码比特序列及其在 LDPC 码字序列中的位置“冻结”,即在此后的 LDPC 的译码迭代中保持不变. 这些信息作为 AEI 传递给 LDPC 译码器的 VN 模块,从而可以利用 RS 获得的正确信息使得概率跟踪更加准确,加快译码收敛速度. 因此, AEI 的可靠性对联合译码性能非常重要.

在文献 [20] 中已证明,假设纠 e 个错误符号的 q 进制 RS 码被用来纠正错误符号数小于等于 γ 的所有错误模式,其中 γ 为小于 e 的非负整数,则纠错后的漏检率的上界为

$$P_u(E, \gamma) < q^{-2e} \sum_{h=0}^{\gamma} \binom{q-1}{h} (q-1)^h. \quad (3)$$

对于本文使用的 GF(2⁶) 上的 RS(30, 26) 码, $q = 64$, $e = 2$, $\gamma = 0, 1$. 则其纠错后的漏检率 $P_u(E)$ 的上界为 $P_u(E) < P_u(E, 1) = 2.5202 \times 10^{-4}$.

通过文献 [21] 可知,对于码长为 n_c 比特,信息位占 k_c 比特, CRC 校验位为 $l_c = n_c - k_c$ 比特的码字,可以表示为多项式形式

$$T(X) = X^{l_c} P(X) + R(X), \quad (4)$$

其中 $P(X)$ 为信息位多项式, $R(X)$ 为 CRC 校验位多项式. 假设 $G(X)$ 是 CRC 生成多项式,传输错误模式为 $E(X)$,则检测到的错误 $E'(X)$ 可被表示为

$$E'(X) = \frac{T(X) + E(X)}{G(X)} = \frac{E(X)}{G(X)}. \quad (5)$$

注意上式为多项式运算, $T(X)$ 为 $G(X)$ 的倍数. 当 $E'(X) = 0$ 时会产生未被检测的错误,并且 $E(X)$ 可以表示成 $\{G(X), X \cdot G(X), X^{k_c} \cdot G(X)\}$ 的组合. 假设码字中所有的比特互相独立且具有相同的错误概率,那么 CRC 的漏检率 $P_u(D)$ 为

$$P_u(D) = \frac{\sum_{i=1}^{k_c+1} \binom{k_c+1}{i}}{\sum_{i=1}^{n_c} \binom{n_c}{i}}. \quad (6)$$

只有在 RS 译码器和 CRC 校验都漏检的情况下,产生的错误码字会被当作正确译码送入 AEI 产生器模块,那么 AEI 产生器模块的漏检率上界为

$$P_u(\text{AEI}) = P_u(E)P_u(D). \quad (7)$$

在联合译码设计中,我们希望 $P_u(\text{AEI})$ 保证级联码达到目标误码率的要求即可. 本文对 RS64(30, 26) 码的信息码元进行 CRC 校验编码,校验位为 11 比特,此时 $k_c = 26 \times 6 = 156$ 比特,对其进行 CRC 编码后 $n_c = k_c + 11 = 167$ 比特. 根据式 (6) 计算得 $P_u(D) = 9.7656 \times 10^{-4}$,由式 (7) 得到 $P_u(\text{AEI}) < 2.46 \times 10^{-7}$; 在我们的仿真实验中可以验证,联合译码可以保证误码率达到 10^{-7} 的水平. 因此 AEI 产生器可以产生高可靠性的 AEI 信息,从而保证联合译码方案的性能提升.

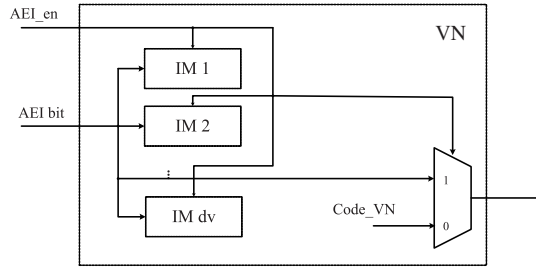


图 5 AEI-VN 结构图

Figure 5 Structure of AEI-VN

3.3 概率 LDPC 译码器变量节点结构设计

概率 LDPC 译码器校验节点的结构采用图 1(a) 所示结构, 不再赘述. 我们对变量节点结构进行了改进, 使其能够处理 AEI 信息. 我们分别使用 TFM 和 counter 作为概率跟踪器. 在传统的概率变量节点结构中, 存储在 IM 中的比特来自与之相连的 CN 的一致输入, 重随机比特流 $r(t)$ 的概率由输入端口的一致信息或者 IM 的存储值计算. 以使用 TFM 的重随机模块为例, 当输入 IM 的比特不一致时, IM 将随机选择存储的比特送入 TFM, 由 TFM 跟踪到的概率为

$$P_i^{t+1} = (1 - \beta)P_i^{t+1} + \beta P_{r(t)}, \quad i = 0, 1, \dots, N - 1, \quad (8)$$

其中 N 为 LDPC 的码长, β 为松弛因子, $r(t)$ 是由 IM 得到的比特流. 然而 IM 只能存储之前少量的一致信息, 这可能会导致跟踪概率不准确和收敛速度变慢. 基于 3.2 小节的分析, 我们加入的 AEI 信息是可靠的, 因此在我们的方案中变量节点需要处理 AEI 信息, 如图 5 所示, AEI 信息包括正确译码的码字 AEI bit 和 AEI 使能位 AEI.en. AEI bit 对应位置的变量节点的 AEI.en 为 1, 否则为 0. 当 AEI.en = 1 时, 变量节点用对应的 AEI bit 替换 IM, 且 IM 停止更新过程, 同时输出的硬判决码字也被正确信息代替, 这一步我们称之为“冻结”. 重随机比特流 $r(t)$ 的概率可以替换为

$$P_{\text{AEI}} = \begin{cases} 1, & \text{AEI bit} = 1, \\ 0, & \text{AEI bit} = 0, \end{cases} \quad (9)$$

则式 (8) 可以改写为

$$P_i^{t+1} = (1 - \beta)P_i^{t+1} + \beta P_{\text{AEI}}, \quad i = 0, 1, \dots, N - 1. \quad (10)$$

这样 TFM 跟踪的概率也将被“冻结”为高置信度的概率, 也就是说已经译码正确的 LDPC 编码位将一直传递正确的信息, 使得变量节点和校验节点之间传递的信息更加可靠. 同理, counter 结构也适用于上述分析.

3.4 算法及复杂度分析

本小节提供了联合迭代译码算法的整体表述, 如算法 1 所示. 其中, LLR 是从信道接收到的软信息, N_0 是估计的噪声功率, C 是级联码中 RS 码的个数, iter 为最大外迭代译码周期. 算法流程具体如下: 概率 LDPC 译码器模块中, L2S 首先利用接收到的 LLR, N_0 将软信息转换为比特流, 比特流信息在 AEI-VNs 和 CNs 之间更新迭代, 每一次信息迭代后利用 LDPC 的校验关系进行校验, 若校验通过则停止译码, 输出译码结果; 否则将通过 Test-vector Generator 产生测试向量送入 RS 译码模块. 若

算法 1 RS-LDPC 联合概率译码算法

输入: LLR, N_0 , C , DC_0 , DC_t , iter;

初始化: $l = 0$, $S = [s_0, s_1, \dots, s_{C-1}] = \mathbf{0}$, $AEI_i = \mathbf{0}$, $AEI_{len}_i = 0, i = 0, 1, \dots, C - 1$;

- 1: **for** it = 0 \rightarrow iter - 1 **do**
- 2: **if** it == 0 **then**
- 3: $DC = DC_0$;
- 4: **else**
- 5: $DC = DC_t$;
- 6: **end if**
- 7: **for** $l = 0 \rightarrow DC - 1$ **do**
- 8: 根据 LLR, N_0 , AEI_i 进行 LDPC 概率译码产生译码码字 y^l 并计算校验和 $sum^{(l)}$;
- 9: **if** $sum^{(l)} == 0$ **then**
- 10: $y = y^l$ 并终止译码过程;
- 11: **end if**
- 12: **end for**
- 13: 将 y 转换为 C 组 RS 码字序列 $R = [r_0, r_1, \dots, r_{C-1}]$;
- 14: **for** $i = 0 \rightarrow C - 1$ **do**
- 15: **if** $s_i == 0$ **then**
- 16: 对 r_i 进行 RS 硬判决译码得到 RS 译码码字 r'_i 并进行 CRC 校验,
- 17: $s_i =$ RS 硬译码器译码成功且通过 CRC 校验?1 : 0;
- 18: **if** $s_i = 1$ **then**
- 19: $AEI_i = r'_i$; $AEI_{len}_i = 1$;
- 20: **end if**
- 21: **end if**
- 22: **end for** \ \ C 组 RS 硬译码器并行运行步骤 14~22.
- 23: **if** $S == \mathbf{1}$ **then**
- 24: $y =$ Interleave[$AEI_0, AEI_1, \dots, AEI_{C-1}$] 并终止译码过程;
- 25: **end if**
- 26: **end for**
- 27: $y = y^l$;

输出: y .

全部 RS 译码器译码成功且通过 CRC 校验, 则终止译码, 输出码字; 否则将进入 AEI Generator 模块进行处理产生 AEI, 然后送回 LDPC 译码模块的 AEI-VN 进行下一次迭代, 直到达到最大外迭代译码周期.

译码延迟方面, 已有研究^[22]表明 RS 硬译码器可以采用流水的方式实现高吞吐率, 因此在我们的联合译码方案中 RS 码的 BM 译码器可以采用流水的方式提高吞吐率, 而概率 LDPC 译码器对于比特顺序不敏感, 从而对性能影响不大. 那么联合迭代译码算法的最大译码延迟与概率 LDPC 译码器的最大译码延迟相等, 即 $DC_0 + DC_t \times (\text{iter} - 1)$ DCs. 相比需要多位宽操作的 BP 和 Chase 算法, 所需的时钟周期呈数量级减小. 在算法 1 中, 我们设置了两个提前终止机制, 即在第 10 行使用了 LDPC 校验功能和在第 14~25 行判别所有 RS 码模块是否译码成功且通过附加的验证过程. 译码时延可作为一个随机变量, 在不同信噪比下, 提前终止机制使得其均值远小于最大译码延迟, 如图 6 所示.

复杂度方面, 由于 LDPC 译码算法的硬件复杂度远大于 RS 译码算法, 因此在 RS-LDPC 级联码的译码算法中复杂度主要由 LDPC 译码算法决定. 我们的概率 LDPC 译码器结构参考了文献 [9] 的模型, 其归一化硬件效率为 16.92 Gb/s/mm^2 (@ $E_b/N_0 = 5.5 \text{ dB}$), 文献 [15] 中 BP 算法的归一化硬件效率为 0.36 Gb/s/mm^2 , 那么相同条件下, 概率 LDPC 译码器的硬件效率是 BP 算法的 47 倍. 另外,

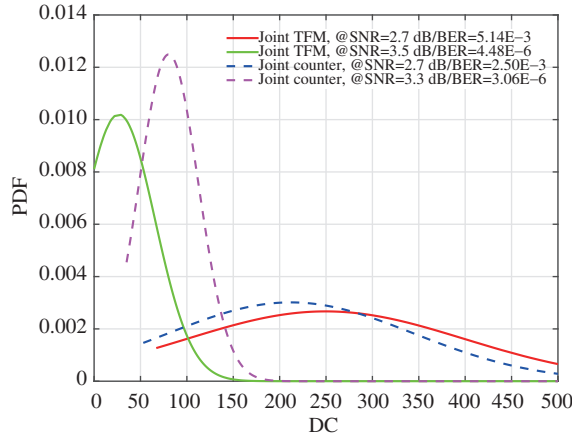


图 6 在 $DC_0 = 50$ DCs, $DC_t = 10$ DCs, $iter = 46$ 下, 基于 TFM 和 counter 的概率联合迭代译码器在不同信噪比下的译码延迟概率分布曲线

Figure 6 The probability distribution curves of decoding latency of the stochastic joint iterative decoder based on TFM and counter under different SNRs when $DC_0 = 50$ DCs, $DC_t = 10$ DCs, $iter = 46$

假设 BP 算法需要用 b 比特量化输入软信息, 那么概率 LDPC 译码器仅需要 1 比特, 连线数是 BP 算法的 $1/b$, 大大简化了 BP-based (包括 MSA) 算法的非线性运算, 减小了硬件布线压力和面积开销. 我们的算法继承了概率 LDPC 译码器的这些优势. 对于 RS 码, BM 硬判决算法的复杂度为 $\vartheta(n^2)$, 其中 n 为 RS 码长; 而 Chase 算法的复杂度为 $\vartheta(n^2 2^\lambda)$ [4], 其中 λ 为最不可靠的比特数, λ 越大性能越好, 但复杂度也呈指数级增长, 是 BM 算法的 2^λ 倍, 显然 BM 算法复杂度更低. 同时算法 1 中步骤 14~22 可以并行进行, 因此联合迭代译码算法的复杂度与 C 个 RS 码 BM 硬判决译码器和 1 个 LDPC 的概率译码器的复杂度之和近似相等.

根据上述分析, 基于 Chase 和 BP 的 RS-LDPC 级联码译码算法硬件实现复杂度极高, 而本文提出的基于概率计算的联合迭代译码算法, 由于使用了概率 LDPC 译码架构和 RS 码的硬译码器, 在保证低复杂度和译码延时的情况下, 通过合理的设计得到了更优的译码性能, 使得算法更具有应用性.

4 仿真结果与分析

在 BPSK 调制下, 对提出的联合迭代译码算法进行仿真. 采用 RS64(30, 26) 码作为外码, IEEE 802.3an 10GBASE-T 标准使用的 (2048, 1723)LDPC 码作为内码. 交织方式采用随机交织, 利用每段 RS 码字的信息位产生 11 比特 CRC 校验位, 由于 RS 码的编码码字不会填满 LDPC 码的信息位, 将所有 CRC 校验位部署在 LDPC 信息位的空余位. 我们同样对其他不同的级联算法和联合算法进行了仿真, 需要注意的是级联译码算法没有外迭代, 也不需添加 CRC. 在基于 FPBP-BM 级联译码算法中, BP 的信息迭代次数为 15; 在基于 TFM 和 counter 的级联译码算法中, 补零位不算入有效码率, 最大译码周期为 500 DCs; 在基于 FPBP-Chase 的联合译码算法中, 内迭代中 BP 的信息迭代次数为 15, Chase 算法参数 λ 设为 6, 最大外迭代次数为 5; 在我们提出的联合译码方案中, 最大译码周期为 500 DCs, $DC_t = 10$ DCs, 由于 DC_0 取值不同, 最大外迭代译码周期 $iter$ 相应改变. 仿真结果如图 7(a) 和 (b) 所示. 在 BER 为 10^{-6} 处, 我们提出的算法使用 counter 结构相比基于 FPBP-BM 级联译码算法可以获得 0.3 dB 左右的增益; 相比基于 FPBP-Chase 联合迭代算法获得了约 0.2 dB 的增益. 使用 TFM 结构虽然性能没有增益, 但是在大大简化硬件复杂度的情况下逼近了软算法的性能. 为了简化硬件复

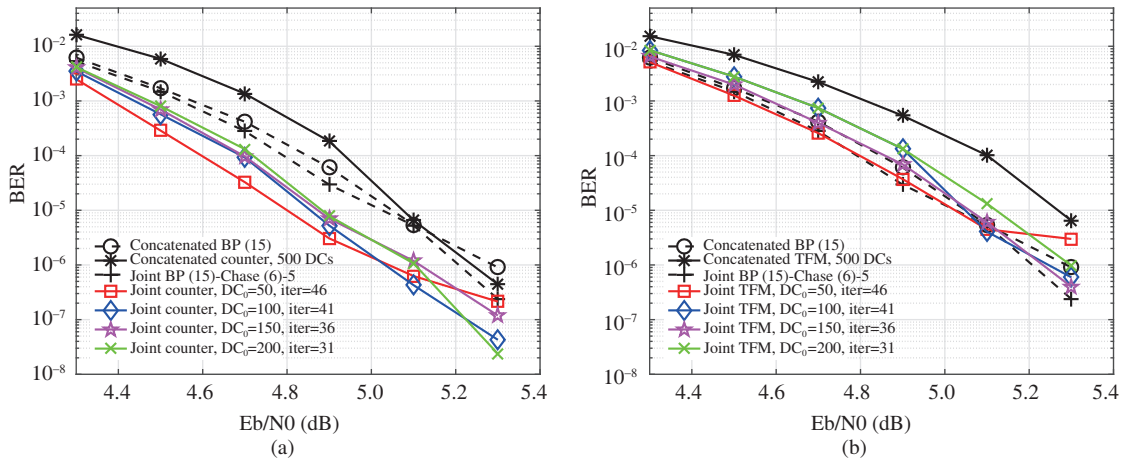


图 7 不同 DC 组合下概率 (a) counter 和 (b) TFM 译码算法译码性能对比图

Figure 7 Comparison of decoding performance of stochastic decoding algorithm based on (a) counter and (b) TFM under different DC combinations

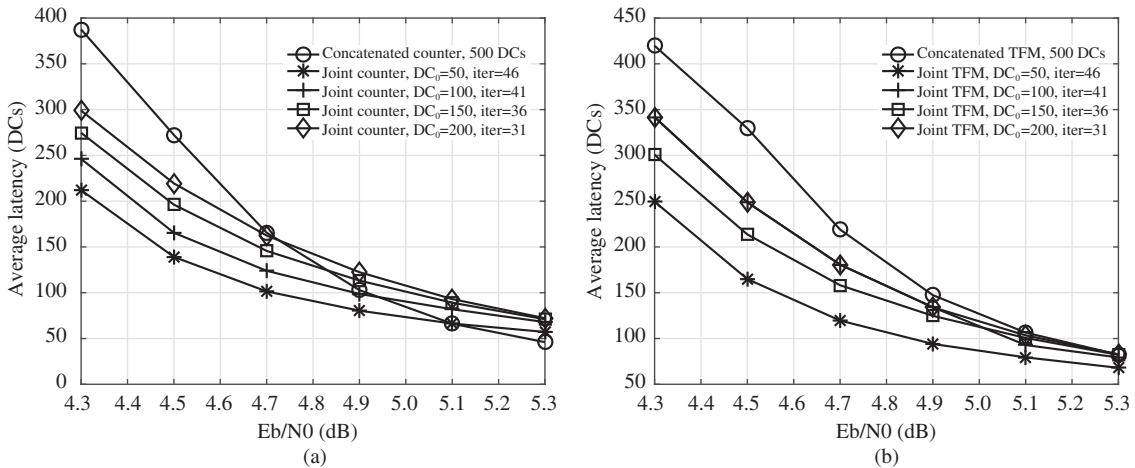


图 8 不同 DC 组合下概率 (a) counter 和 (b) TFM 译码算法平均译码延迟对比图

Figure 8 Comparison of average decoding latency of stochastic decoding algorithm based on (a) counter and (b) TFM under different DC combinations

杂度, 概率 LDPC 译码器会有一些性能损失, 但是我们提出的概率联合迭代译码算法弥补了性能损失并获得了更多的增益. 另外, 在最大译码周期相同的情况下, 我们提出的联合译码算法与同等条件的级联译码算法相比最多减少了约一半的译码延迟, 如图 8(a) 和 (b) 所示.

我们还探究了在最大总译码周期和 DC_t 相同的情况下, 不同的 DC_0 对译码性能的影响, 分别仿真了 $DC_t = 10$ DCs, 最大总译码周期都为 500 DCs 的条件下, $DC_0 = 50, 100, 150, 200$ DCs 的性能. 在 BER 为 10^{-6} 之前, $DC_0 = 50$ DCs 具有最好的性能, 且平均译码延迟也最低, 这意味着更高的吞吐量; 但是在较高信噪比处会出现错误平层现象. 而其他 3 种情况在高信噪比下并未出现平层现象, 表现出更好的性能, 并且高信噪比下平均译码延迟相差不大. 因此可以根据实际需求选择合适的参数以获得性能和硬件效率的折中. 另外, 我们统计了在 $DC_0 = 50$ DCs, $DC_t = 10$ DCs, iter = 46 条件下, 基于两种结构的概率联合迭代译码器分别在高低信噪比下的译码延迟概率分布, 如图 6 所示. 可以看

出, 随信噪比增加, 曲线的平均值和方差均减小, 表明随着信噪比的提高, 平均译码延迟减小且远低于最大译码延迟. 在相同信噪比下, 基于 counter 的联合迭代译码器的译码延迟曲线方差比基于 TFM 的联合迭代译码器的译码延迟曲线方差小, 说明基于 counter 的联合迭代译码器译码延迟更加平稳.

5 结论

我们提出了一种新的概率 RS-LDPC 联合译码方案. 在我们的算法设计中, 采用了一种 LDPC-RS 外迭代的信息更新方式, 根据 RS 译码器的输出结果产生附加外信息 (AEI) 传递回概率 LDPC 译码器, 可以改善 LDPC 的信息更新性能, 加快译码收敛速度; 在高信噪比下, 平均 50 个译码周期即可完成译码. 同时, 我们利用概率 LDPC 译码器的输出结果产生 RS 译码器的测试向量, 使得 RS 译码器可以以 BM 硬译码器的复杂度逼近软译码器的性能, 显著提高译码性能. 仿真结果表明, 与基于浮点 BP 和 BM 的 RS-LDPC 级联译码相比, 我们提出的联合迭代译码方案可以获得 0.3 dB 的译码增益. 另外, 我们的方案继承了概率 LDPC 译码算法高硬件效率的优点, 相比基于浮点 BP 和 Chase 的译码算法, 大大降低了硬件实现复杂度; 并且 RS 译码模块可以并行进行, 从而使算法能够全并行实现, 实现了高吞吐率, 具有重要的研究和应用价值.

参考文献

- 1 Forney G D. Concatenated Codes. Cambridge: MIT Press, 1966
- 2 Chase D. Class of algorithms for decoding block codes with channel measurement information. *IEEE Trans Inform Theor*, 1972, 18: 170–182
- 3 Leroux C, Hemati S, Mannor S, et al. Stochastic chase decoding of Reed-Solomon codes. *IEEE Commun Lett*, 2010, 14: 863–865
- 4 Mani H, Hemati S. Symbol-level stochastic chase decoding of Reed-Solomon and BCH codes. *IEEE Trans Commun*, 2019, 67: 5241–5252
- 5 Fossorier M P C, Mihaljevic M, Imai H. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. *IEEE Trans Commun*, 1999, 47: 673–680
- 6 Gaudet V C, Rapley AC. Iterative decoding using stochastic computation. *Electron Lett*, 2003, 39: 299
- 7 Tehrani S S, Gross W J, Mannor S. Stochastic decoding of LDPC codes. *IEEE Commun Lett*, 2006, 10: 716–718
- 8 Gaudet V C, Rapley A C. Iterative decoding using stochastic computation. *Electron Lett*, 2003, 39: 299–301
- 9 Ueng Y L, Wang C Y, Li M R. An efficient combined bit-flipping and stochastic LDPC decoder using improved probability tracers. *IEEE Trans Signal Process*, 2017, 65: 5368–5380
- 10 Naderi A, Mannor S, Sawan M, et al. Delayed stochastic decoding of LDPC codes. *IEEE Trans Signal Process*, 2011, 59: 5617–5626
- 11 Shi Z, Zhang Z, Li S. Joint iterative decoding for LDPC codes and RS codes. *J Univ Electr Sci Technol China*, 2006, 35: 657–660
- 12 Qiu J, Liu S, Chen L. Concatenated reed-solomon/spatially coupled LDPC codes. In: *Proceedings of International Conference on Wireless Communications and Signal Processing (WCSP)*, 2019
- 13 Tehrani S S, Mannor S, Gross W J. Fully parallel stochastic LDPC decoders. *IEEE Trans Signal Process*, 2008, 56: 5692–5703
- 14 Tehrani S S, Naderi A, Kamendje G A, et al. Tracking forecast memories in stochastic decoders. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, 2009
- 15 Tehrani S S, Naderi A, Kamendje G A, et al. Majority-based tracking forecast memories for stochastic LDPC decoding. *IEEE Trans Signal Process*, 2010, 58: 4883–4896
- 16 Chen J, Hu J. Sliding window method for stochastic LDPC decoder. In: *Proceedings of IEEE International Symposium on Circuits & Systems*, 2011

- 17 Zhang Q, Chen Y, Wu D, et al. An area-efficient architecture for stochastic LDPC decoder. In: Proceedings of IEEE International Conference on Digital Signal Processing (DSP), 2015
- 18 Zhang Q, Chen Y, Wu D, et al. Convergence-optimized variable node structure for stochastic LDPC decoder. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016
- 19 Wu D, Chen Y, Zhang Q, et al. Latency-optimized stochastic LDPC decoder for high-throughput applications. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), 2015. 3044–3047
- 20 Lin S, Costello J D. Error Control Coding. 2nd ed. Beijing: China Machine Press, 2007
- 21 Khamy M E, Lee J, Kang I. Detection analysis of CRC-assisted decoding. IEEE Commun Lett, 2015, 19: 483–486
- 22 Mondal A, Thatimattala S, Yalamaddi V K, et al. Efficient coding architectures for Reed-Solomon and low-density parity-check decoders for magnetic and other data storage systems. IEEE Trans Magn, 2018, 54: 1–15

RS-LDPC stochastic joint decoding algorithm

Yining SUN*, Qiu HUANG, Jianhao HU* & Kai KANG

National Key Laboratory of Science and Technology on Communication, University of Electronic Science and Technology of China, Chengdu 611731, China

* Corresponding author. E-mail: ynsun3842@163.com, jhhu@uestc.edu.cn

Abstract Due to complexity issues, low-density parity-check (LDPC) codes often use a partial parallel decoding architecture, which cannot easily meet the requirements of high-speed and ultra-high-speed applications. Conversely, the Reed-Solomon (RS) codes often use hard decision decoding algorithms in existing decoding schemes. Therefore, there is room for further optimization in the complexity and performance of concatenated RS-LDPC decoding. In this paper, we propose an RS-LDPC stochastic joint iterative decoding algorithm based on the principle of stochastic computing. This algorithm inherits the advantages of the low complexity of stochastic LDPC decoders and the feature of stochastic RS decoding algorithms, which can use hard decoders. Meanwhile, the decoding performance is improved through the joint iterative decoding architecture. In this respect, we design a test-vector generation method based on LDPC probability values and adopt a novel additional extrinsic information generation mechanism. Such a mechanism is applied to the structure of the variable nodes of the stochastic LDPC decoder. The simulation results show that at a bit error rate of 10^{-6} , the proposed algorithm can obtain a gain of 0.3 dB as compared with the RS-LDPC cascaded decoding scheme using floating-point belief propagation and Berlekamp-Massey hard decision decoding algorithm. Its implementation complexity is the complexity of a stochastic LDPC decoder and several RS hard decoders. The decoding latency is determined by the latency of the stochastic LDPC decoder. Therefore, the joint iterative decoding algorithm proposed in this paper has a very good application value.

Keywords RS codes, LDPC codes, stochastic computing, joint iterative decoding