

基于容错高斯消元的线性分组码校验矩阵重建

王忠勇¹, 张洪², 巩克现¹, 王伟¹, 张玲¹, 朱俊昊¹

(1. 郑州大学电气与信息工程学院, 河南 郑州 450001; 2. 郑州大学计算机与人工智能学院, 河南 郑州 450001)

摘要: 针对线性分组码校验矩阵重建容错性能差的问题, 提出了一种基于容错高斯消元的校验矩阵重建算法。首先, 通过多次随机抽取码字的部分比特构建残缺码字, 依据比特对数似然比绝对值的均值和最小值计算其可靠度, 从而筛选出高可靠度的残缺码组。其次, 针对残缺码组基的部分错误状态, 利用容错高斯消元算法求解与之对应的疑似校验向量, 进而从中判定并还原出真实的校验向量。最后, 依据码字比特能否通过已知校验向量更新比特对数似然比绝对值, 进一步提升比特可靠度, 为下次随机抽取部分比特带来增益。仿真结果表明, 在相同识别条件下, 所提算法的重建率高于现有算法。对于IEEE802.11n协议下的LDPC(648,324)码, 现有算法在误码率达到0.005 5时即失效, 而所提算法在误码率为0.008时, 校验矩阵重建率仍能达到98%以上。

关键词: 线性分组码; 校验矩阵重建; 码字筛选; 高斯消元

中图分类号: TN911.7

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2025065

Reconstruction of linear block code parity-check matrix based on fault-tolerant Gaussian elimination

WANG Zhongyong¹, ZHANG Hong², GONG Kexian¹, WANG Wei¹, ZHANG Ling¹, ZHU Junhao¹

1. School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou 450001, China

2. School of Computer Science and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China

Abstract: Aiming at the problem of poor fault-tolerant performance in the reconstruction of the parity-check matrix of linear block codes, a parity-check matrix reconstruction algorithm based on fault-tolerant Gaussian elimination was proposed. Firstly, partial bits of codewords were randomly sampled multiple times to construct incomplete codewords. The reliability was calculated based on the mean and minimum of the absolute values of the bit log-likelihood ratios, so as to screen out high-reliability incomplete code groups. Secondly, for the partially incorrect states of the basis of the incomplete code groups, the fault-tolerant Gaussian elimination algorithm was used to solve the corresponding suspected parity-check vectors, and then the true parity-check vectors were determined and restored from them. Finally, according to whether the codeword bits could pass through the known parity-check vectors, the absolute values of the bit log-likelihood ratios were updated to further improve the bit reliability, bringing gains for the next random sampling of partial bits. The simulation results showed that, under the same identification conditions, the reconstruction rate of the proposed algorithm is higher than that of the existing algorithms. For the LDPC(648,324) code under the IEEE802.11n protocol, the existing algorithms fail when the bit error rate reaches 0.005 5, while the reconstruction rate of the proposed algorithm for the parity-check matrix still reaches over 98% when the bit error rate is 0.008.

Keywords: linear block code, parity-check matrix reconstruction, codeword selection, Gaussian elimination

收稿日期: 2024-11-26; 修回日期: 2025-03-26

通信作者: 张玲, zhangling@cyberyian.com

基金项目: 国家自然科学基金资助项目(No.61901417); 河南省科技攻关计划基金资助项目(No.242102211107)

Foundation Items: The National Natural Science Foundation of China (No.61901417), The Key Scientific and Technological Project of Henan Province (No.242102211107)

0 引言

信道编码识别是指仅通过分析截获的信息序列,识别其中所采用的编码方式和编码参数(如码长、起点和校验矩阵等),从而提取信息序列中所承载的通信内容和目标属性等信息。线性分组码是一种在信道编码中广泛使用的重要编码方式,包括常见的BCH(Bose-Chaudhuri-Hocquenghem)码、RS(Reed-Solomon)码、LDPC(low-density parity-check)码、汉明码等。线性分组码的识别是信道编码识别的核心组成部分,广泛应用于智能通信、军事侦察和信息对抗等领域,具有重要的现实意义^[1]。

线性分组码的识别可以通过分析特定码字类型的编码结构来实现^[2-6],但该类方法仅适用于特定的编码类型。还有一类通用方法,通过分析线性分组码元之间的线性关系来实现。张永光等^[7]通过构建码字矩阵进行直接高斯消元处理,以码字矩阵的秩亏情况来确定码长,同时根据移位矩阵的秩亏情况来确定起点,进而得出生成矩阵和校验矩阵。然而,随着误比特率的增加,该方法的有效性迅速下降。吴昭军等^[8]针对LDPC码稀疏校验矩阵的重建问题,首先采用随机抽取码字的部分比特构建残缺码字,再从中随机选择部分残缺码字进行直接高斯消元处理,从生成的低维对偶空间中寻找稀疏校验向量。该方法降低了计算复杂度并提升了容错性。然而,由于残缺码字的选择是随机的,未能充分利用码组中的高质量码字,因此在高误码率条件下,重建性能仍有提升空间。刘恒燕等^[9]在随机抽取码字的基础上,利用改进的归一化最小和(NMS, normalized minimum sum)译码方法改善码组整体的质量,进一步提高了重建率。王忠勇等^[10]采用剔除错误码字和翻转最不可靠比特的方法实现码组整体质量提升。张天骐等^[11]采用剔除错误码字和改进分层置信传播(LBP, layered belief propagation)译码的方法实现码组整体质量提升。然而,上述这些方法只能在部分校验向量已知的情况下提升码组质量,且译码方法并不适用于所有线性分组码。此外,码字剔除不够谨慎,未考虑到一个校验关系被破坏并不代表所有校验关系被破坏。直接高斯消元算法的行叠加操作会导致错误传播到其他码字,进一步减少了可用校验向量的数量。包昕等^[12]采用 k 阶列消元方法求解校验向量,并利用剔除错误码字的方法提高码组内无误码码字的比例,

将有误码情况下的识别问题转化为无误码的情况。但该方法容错性能有限,且所需码字数量巨大。刘杰等^[13]利用高斯列消元算法对码字矩阵进行处理,并设定相关列的判决门限,最终依据相关列归一化的数值识别码长和起点,但该方法在码长增大时失效。由于上述利用高斯消元法求解校验向量的方法仅针对未被破坏的校验关系,因此得到的校验向量的数量是有限的。游凌等^[14]和杨晓炜等^[15]利用Walsh变换求解校验向量。该方法在容错性方面表现良好,但是在码长较大时,计算量会急剧增加。王垚等^[16]利用码字软解调序列的最小值筛选码字,但因为最小值不能全面反映码字的质量,故筛选效果有待提高。

以上分析表明,高误码率下传统算法的校验矩阵重建率低主要有2个原因:一是未能充分利用码组中的高质量码字,其随机选择部分码字进行识别,增加了重建的难度;二是所使用的高斯消元算法容错性低。传统算法主要利用直接高斯消元与列高斯消元求解疑似校验向量,直接高斯消元含初等行列变换,其中行叠加操作(码字与码字叠加)会使码字间误码扩散。列高斯消元虽无行叠加操作,却仍无法求解被破坏的校验关系,容错性能有限。

为此,本文提出了一种基于容错高斯消元的线性分组码校验矩阵重建算法。首先,该算法通过多次随机抽取码字的部分比特构建残缺码字,依据比特对数似然比(LLR, log-likelihood ratio)绝对值的均值和最小值计算残缺码字可靠度,并筛选出高可靠度的残缺码组,从而充分利用误码组中未受损的校验关系。其次,针对残缺码组基的部分错误状态,利用容错高斯消元算法求解与之对应的疑似校验向量,使其涵盖更多真实的校验向量,从而增强高斯消元的容错能力。最后,利用已知校验向量校验码字,并根据码字校验情况持续更新比特的对数似然比,进一步提高码字可靠度,从而加速校验矩阵重建并提升重建率。

1 问题描述与分析

线性分组码校验矩阵重建的前提是码长和起点已知,文献[13]对此问题已有研究。因此,本文在假设已知码长 n 和起点 sp 的条件下,探讨校验矩阵 H 的重建方法。设接收到 (n,k) 线性分组码的软解

调序列为 $\mathbf{r} = [r_1, r_2, \dots, r_L]$, 硬判决比特序列为 $\mathbf{y} = [y_1, y_2, \dots, y_L]$, 其中 $y_i \in \{0, 1\}$, L 为序列长度。先将硬判决比特序列 \mathbf{y} 的前 $\text{sp} - 1$ 比特删除, 再将剩余比特分成 $N = \lfloor \frac{L - \text{sp} + 1}{n} \rfloor$ 组, 可得 $N \times n$ 的码字矩阵 \mathbf{Y} 为

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \\ y_{n+1} & y_{n+2} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{(N-1)n+1} & y_{(N-1)n+2} & \cdots & y_{Nn} \end{bmatrix} \quad (1)$$

其中, \mathbf{y}_i 为完整码字。按照同样方式将软解调序列 \mathbf{r} 排成矩阵 \mathbf{R} , 如式(2)所示。

$$\mathbf{R} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & \cdots & r_n \\ r_{n+1} & r_{n+2} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{(N-1)n+1} & r_{(N-1)n+2} & \cdots & r_{Nn} \end{bmatrix} \quad (2)$$

其中, 每个元素都是 \mathbf{Y} 相同位置比特的软解调信息。

在无误比特情况下, \mathbf{Y} 中的每个码字都满足 $\mathbf{H}\mathbf{y}_i^T = \mathbf{0}$ 的线性约束, 可直接对 \mathbf{Y} 进行高斯消元求出码字的全部校验向量, 实现校验矩阵重建。然而在有误比特情况下, 直接高斯消元只能得到部分校验向量, 而且校验向量的个数随着误比特率的增加逐渐减小, 直至为零。文献[8]利用 LDPC 校验向量稀疏的特点, 通过随机抽取码字的部分比特构建残缺码字, 在低维度下求解校验向量, 减少了求解校验向量的计算复杂度并提高了容错性。残缺码字能够求解部分校验向量的前提是这些校验向量的校验位在残缺码字中保持完整。因此, 无论线性分组码的校验矩阵是否稀疏, 只要残缺码字中存在一些完整的校验位, 即可用来求解部分校验向量。

文献[8]在构建残缺码字矩阵后, 随机抽取部分行向量形成方阵, 再对该方阵进行高斯消元以求解对偶向量。由于方阵的行向量是随机抽取的, 其码字质量与残缺码字矩阵的整体质量相当, 未能充分利用矩阵中的高质量残缺码字。虽然文献[9-11]采取剔除错误码字、翻转不可靠比特、译码等方法提高了残缺码字矩阵的整体质量, 但是仍以随机抽取行来构造方阵, 而且运用这些方法的前提是已经获得部分校验向量。本文算法利用比特对数似然比绝对值的均值和最小值计算残缺码字的可靠度, 并据此筛选残缺码字, 不需要校验向量即可筛选出高

质量残缺码字。

传统高斯消元法求解校验向量仅适用于校验关系未被破坏的情况。本文提出的容错高斯消元算法能够在校验关系遭到破坏的条件下求解校验向量。该算法首先通过控制高斯消元的过程和结果, 对于筛选出的残缺码组基, 直接求解其无错和有单个错误码字时的疑似校验向量。然后, 利用这些已知的疑似校验向量, 推导出更多错误状态对应的疑似校验向量, 从而求解码字中被破坏的校验关系, 提高获得真实校验向量的概率。此外, 为了进一步提高码字可靠度, 本文算法通过已知校验向量更新了比特对数似然比, 为残缺码字再次筛选提供更可靠的依据。具体而言, 对参与且通过校验的比特, 将其对数似然比绝对值加上调节因子 α 。对参与但未通过校验的比特, 将其对数似然比绝对值减去调节因子 α , 其经验值为 1。此方法将校验向量中的可靠信息融入比特对数似然比绝对值, 使计算码字可靠度时仅需利用比特对数似然比绝对值即可。

综上所述, 基于容错高斯消元的线性分组码校验矩阵重建流程如图 1 所示。文献[8]已详细分析了随机比特抽取过程, 码字比特对数似然比的更新也已在本节阐述。接下来, 将在重建算法中重点分析码字筛选策略、容错高斯消元法求解疑似校验向量的过程以及疑似校验向量的判决方法。

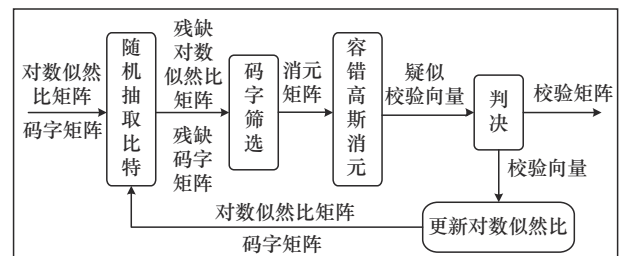


图 1 基于容错高斯消元的线性分组码校验矩阵重建流程

2 校验矩阵重建算法

本节首先从码字筛选的 2 个关键步骤展开分析, 即码字筛选数量的确定和码字可靠度的计算。然后详细分析基于容错高斯消元的疑似校验向量求解过程, 接着讨论疑似校验向量的判决方法。最终, 综合这些步骤形成完整的校验矩阵重建算法。

2.1 码字筛选数量的确定

假设筛选出 K 个码字向量以构成码字向量组, 该向量组的作用是获取校验向量。因此, 对于残缺

码字向量组,应包含残缺码字空间的一组基。对于完整码字向量组,则需包含完整码字空间的一组基。换言之,码字向量组应具有满秩特性。

设 V_i 表示含有 i 个 GF(2) 上 n 维向量的向量组, $\text{rank}(V_i)$ 表示向量组 V_i 的秩, $V_{i,j}$ 表示含有 i 个 GF(2) 上 n 维向量且秩为 j 的向量组。 $P_{i,j}$ 表示从 n 维线性空间 $V(2^n)$ 中随机选取 i 个向量构成向量组 V_i , 且其秩为 j 的概率。从 $V(2^n)$ 中随机选取一个向量 v , 若 $V_{i,j}$ 已知, 则 v 与 $V_{i,j}$ 的关系只有 v 能或不能被 $V_{i,j}$ 中的向量线性表出 2 种情况。若 v 能够被 $V_{i,j}$ 中的向量线性表出, 则 v 与 $V_{i,j}$ 组成新向量组 $V_{i+1,j}$, 否则组成新向量组 $V_{i+1,j+1}$ 。向量组的状态转移如图 2 所示。

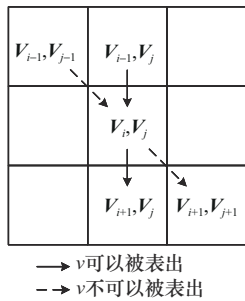


图2 向量组的状态转移

若向量组 $V_{i-1,j}$ 和 $V_{i-1,j-1}$ 已知, 从 $V(2^n)$ 中随机取一个向量 v , 则 $V_{i,j}$ 可由 $\{v, V_{i-1,j}\}$ 和 $\{v, V_{i-1,j-1}\}$ 得到, 如图 2 所示。 $V(2^n)$ 中共有 2^n 个向量, 其中 $2^{\text{rank}(V_{i,j})}$ 个向量可以被 $V_{i,j}$ 线性表出, $2^n - 2^{\text{rank}(V_{i,j})}$ 个向量不可以被线性表出。若 $P_{i-1,j}$ 和 $P_{i-1,j-1}$ 已知, 则可得

$$\begin{aligned}
 P_{i,0} &= P_{i-1,0} \frac{2^0}{2^n}, \quad j=0 \\
 P_{i,j} &= P_{i-1,j} \frac{2^{\text{rank}(V_{i-1,j})}}{2^n} + \\
 &P_{i-1,j-1} \frac{2^n - 2^{\text{rank}(V_{i-1,j-1})}}{2^n}, \quad 0 < j \leq n \quad (3)
 \end{aligned}$$

其中, $\text{rank}(V_{i-1,j}) = 2^j$, $\text{rank}(V_{i-1,j-1}) = 2^{j-1}$ 。设不含向量的空组秩为 0, 则 V_0 的秩为 j 的概率为

$$\begin{aligned}
 P_{0,0} &= 1, \quad j=0 \\
 P_{0,j} &= 0, \quad 0 < j \leq n \quad (4)
 \end{aligned}$$

在已知递推式(3)和边界条件式(4)的条件下, 从 $V(2^n)$ 中随机选取 $K = n + x$ 个向量构成向量组, 计算其满秩的概率, 如图 3 所示。

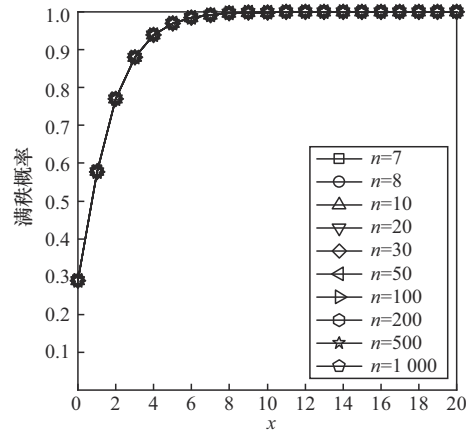


图3 向量组满秩概率

由图 3 可知, 从 $V(2^n)$ 中随机选取 $n + 10$ 个向量, 有大于 0.999 的概率使向量组秩为 n 。因为 (n, k) 线性分组码码字空间是 $V(2^n)$ 的 k 维子空间, 所以在 $K = k + 10$ 的条件下, K 个随机码字有大于 0.999 的概率包含 (n, k) 线性分组码码字空间的一组基。在实际应用中, 虽然 k 的值未知, 但 k 小于 n , 所以可以将 K 设为 $n + 10$ 。

2.2 码字可靠度的计算

考虑在噪声功率为 σ^2 的加性白高斯噪声 (AWGN, additive white Gaussian noise) 信道模型下, 调制方式为二进制相移键控 (BPSK, binary phase-shift keying), 发送序列为 $c = [c_1, c_2, \dots, c_L]$, 其中 $c_i \in \{+1, -1\}$, 则每个硬判决比特 y_i 的后验概率对数似然比 LLR(y_i) 为

$$\text{LLR}(y_i) = \ln \left(\frac{P(c_i = +1 | r_i)}{P(c_i = -1 | r_i)} \right) = \frac{2}{\sigma^2} r_i \quad (5)$$

其中, r_i 表示软解调序列 r 的元素。 $\text{sgn}(\text{LLR}(y_i))$ 表示 y_i 的极性, $|\text{LLR}(y_i)|$ 的大小表示 y_i 取值可信度的大小, $|\text{LLR}(y_i)|$ 的值越大, 则 y_i 取值出错的概率越小。

Y 的行向量 y_i 是码字, 定义 $y_{i,j}$ 表示 y_i 的第 j 个比特。 R 的行向量为 r_i , 定义 $r_{i,j}$ 是向量 r_i 的第 j 个比特。定义 δ_{\min} 和 δ_{mean} 分别表示某个码字中比特对

数似然比绝对值的最小值和平均值, 如式(6)和式(7)所示。

$$\delta_{\min} = \frac{2}{\sigma^2} \min(|r_{i,0}|, \dots, |r_{i,n}|) \quad (6)$$

$$\delta_{\text{mean}} = \frac{2}{\sigma^2} \text{mean}(|r_{i,0}|, \dots, |r_{i,n}|) \quad (7)$$

δ_{\min} 反映了该码字可靠度的下限, δ_{mean} 反映了该码字可靠度的整体趋势。二者分别从不同角度展现了码字的可靠程度。因此, 在评估码字可靠度时, 应综合考虑 δ_{\min} 与 δ_{mean} , 单独依赖其一容易导致码字可靠度降低。设 δ_i 表示码字可靠度, 其计算方式为

$$\delta_i = \delta_{\min} \beta + \delta_{\text{mean}} (1 - \beta) \quad (8)$$

其中, 权重因子 β 的经验值为 0.66。因为高可靠度的码字相较于低可靠度的码字包含误比特的可能性更低, 所以计算出码字的可靠度后, 码字筛选过程就是比较码字可靠度并选择最高可靠度码字的过程。码字筛选算法如算法 1 所示。

算法 1 码字筛选

输入 Y, R, K

输出 已排序的 Y

- 1) 初始化 δ 为零向量
- 2) for $i = 1, 2, \dots, N$ do
- 3) 根据式(6)和式(7)计算 δ_{\min} 和 δ_{mean}
- 4) 根据式(8)计算 δ_i
- 5) end for
- 6) left $\leftarrow 1$, right $\leftarrow N$
- 7) while true
- 8) $j \leftarrow$ left, $k \leftarrow$ right
- 9) $i \leftarrow$ left, pivot $\leftarrow \delta_{\text{left}}$
- 10) while $i \leq k$
- 11) if pivot $< \delta_i$ then
- 12) 交换 δ_i 和 δ_j , 交换 y_i 和 y_j
- 13) $i \leftarrow i + 1$, $j \leftarrow j + 1$
- 14) else if $\delta_i < \text{pivot}$ then
- 15) 交换 δ_i 和 δ_k , 交换 y_i 和 y_k
- 16) $k \leftarrow k - 1$
- 17) else $i \leftarrow i + 1$
- 18) end if
- 19) end while
- 20) if $K < j$ then

- 21) right $\leftarrow j - 1$
- 22) else if $k < K$ then
- 23) left $\leftarrow k + 1$
- 24) else break
- 25) end if
- 26) end while

27) 对 Y 的前 K 行码字根据 δ 从大到小排序

在算法 1 中, 步骤 1) 至步骤 5) 遍历 $N \times n$ 维的 R , 计算复杂度为 $O(nN)$; 步骤 6) 至步骤 26) 在 Y 的行维上线性分组, 计算复杂度为 $O(N)$; 步骤 27) 对 Y 的前 K 行码字进行排序, 计算复杂度为 $O(K \log K)$ 。故算法 1 的总计算复杂度为 $O(nN)$, 其中 N, n 分别为 Y 的行、列维度。

2.3 基于容错高斯消元的疑似校验向量获取

在对 Y 排序后, 取其前 K 行作为消元矩阵 $A_{K \times n}$ 。通过行交换、列交换和列叠加操作, 对消元矩阵 $A_{K \times n}$ 消元, 可得 $A'_{K \times n}$ 为

$$A'_{K \times n} = E_{K \times K} A_{K \times n} F_{n \times n} = \begin{bmatrix} I_{k' \times k'} & \mathbf{0}_{k' \times (n-k')} \\ \mathbf{B}_{(K-k') \times k'} & \mathbf{0}_{(K-k') \times (n-k')} \end{bmatrix} \quad (9)$$

其中, $I_{k' \times k'}$ 是单位阵, $\mathbf{0}_{k' \times (n-k')}$ 和 $\mathbf{0}_{(K-k') \times (n-k')}$ 是全零矩阵, $\mathbf{B}_{(K-k') \times k'}$ 的元素是 0 或 1, $E_{K \times K}$ 是初等行交换矩阵, $F_{n \times n} = [f^1, \dots, f^{n-1}, f^n] = [\Gamma_{n \times k} \Theta_{n \times (n-k)}]$ 是初等列交换矩阵。

$F_{n \times n}$ 包含对 $A_{K \times n}$ 的列叠加和列交换 2 个操作, 列叠加操作可使 $A'_{K \times n}$ 列向量的首非零元位置互异, 且每个首非零元所在的行中仅有首非零元一个非零元。列交换操作则使 $A'_{K \times n}$ 的列向量呈阶梯形。 $E_{K \times K}$ 仅包含对 $A_{K \times n}$ 的行交换操作, 使 $A'_{K \times n}$ 的行向量也呈阶梯形。当 $A'_{K \times n}$ 的行列都呈阶梯形时, 它的左上角是单位阵。因此, 通过行交换、列交换和列叠加操作, 不仅可以使 $A_{K \times n}$ 转换为式(9)所示的形式, 还能避免行叠加所带来的误码扩散问题, 即防止错误码字与其他码字叠加后产生更为严重的错误码字。

设 $E_{K \times K} A_{K \times n} = [y'_1, y'_2, \dots, y'_k]^T$, 其中 $y'_i (1 \leq i \leq K)$ 表示 $E_{K \times K} A_{K \times n}$ 的行向量。 $E_{K \times K} A_{K \times n}$ 的意义是交换 $A_{K \times n}$ 的行, 使 $A_{K \times n}$ 的前 k' 行是其行向量的一组基, 即 $\{y'_1, y'_2, \dots, y'_k\}$ 是 $E_{K \times K} A_{K \times n}$ 和 $A_{K \times n}$ 行向量的一组基, 称为码组基。设 $a'_{ij} (1 \leq i \leq K, 1 \leq$

$j \leq n$)为 $A'_{K \times n}$ 的元素,根据式(9)可知, $A'_{K \times n}$ 又可表示为 $E_{K \times K} A_{K \times n} F_{n \times n}$ 。由矩阵乘法的定义可知, a'_{ij} 是 y'_i 与 f^j 内积的结果。若 $a'_{ij} = 0$,则 f^j 是 y'_i 的对偶向量;若 $a'_{ij} = 1$,则 f^j 不是 y'_i 的对偶向量。码组基与 $F_{n \times n}$ 的乘积结果为 $I_{k' \times k'}$ 和 $0_{k' \times (n-k')}$ 。因此, $\Theta_{n \times (n-k')}$ 的列向量与码组基的全部码字正交, $\Gamma_{n \times k'}$ 的列向量仅与码组基中的一个码字不正交。当 $A_{K \times n}$ 中没有错误比特时, $k' = k$,码组基中没有错误码字。此时, $\Theta_{n \times (n-k')}$ 的列向量与码组基的所有码字正交,就是码字的校验向量。而当 $A_{K \times n}$ 中存在错误比特时, $k' > k$,码组基中有一些码字是错误码字。这些错误码字的部分校验关系被破坏,导致校验向量与码组基中所有码字的内积不再严格全为零。在此情况下,因为 $\Theta_{n \times (n-k')}$ 和 $\Gamma_{n \times k'}$ 的列向量是码组基中大部分码字的对偶向量,所以这些列向量是疑似校验向量。

若将 $\{y'_1, y'_2, \dots, y'_{k'}\}$ 视为包含错误码字的码组基,且其错误码字数量为 n_e ,则每个错误状态对应从 k' 个码字中随机选取 n_e 个错误码字作为错误码字的不同组合,其组合数量可表示为 $C_{k'}^{n_e}$ 。若某个向量与码组基的 n_e 个错误码字不正交,而与其错误余码字正交,则可将其视为在码组基的错误码字数量为 n_e 时某个错误状态下的对偶向量。显然, $\Theta_{n \times (n-k')}$ 的列向量是在 $n_e = 0$ 时的对偶向量, $\Gamma_{n \times k'}$ 的列向量是在 $n_e = 1$ 时的对偶向量。

考虑真实校验向量与码组基中码字的内积结果,其中值为1的错误码字数(即校验不通过的数量)显然会随着误比特率的升高而增加。因此,可以进一步假设码组基中错误码字数 $n_e \geq 2$,从而获得更多的疑似校验向量。

当 $n_e = 2$ 时,码组基的错误状态有 $C_{k'}^2$ 种。定义向量 $f = f^i + f^j$,其中 f^i 和 f^j 取自 $\Gamma_{n \times k'}$ 且 $(i \neq j)$ 。定义向量 d 为码组基与 f 的内积结果,显然向量 d 是码组基与 f^i 和 f^j 内积结果之和,即 $I_{k' \times k'}$ 第 i 和 j 列列向量之和。因此,向量 d 中有2个1, f 是在 $n_e = 2$ 时某个错误状态下的对偶向量。由于 $F_{n \times n}$ 秩为 n ,其列向量构成一个线性无关组。因此,作为 $F_{n \times n}$ 子矩阵 $\Gamma_{n \times k'}$ 的列向量同样构成一个线性无关组。从 $\Gamma_{n \times k'}$ 的列向量中选取2个向量,共 $C_{k'}^2$ 种组合方式,与码组基的 $C_{k'}^2$ 种错误状态一一对应。这

些组合所得向量是在 $n_e = 2$ 时的对偶向量,也是码字的疑似校验向量。以此类推,从 $\Gamma_{n \times k'}$ 的 k' 个列向量中选取 $3, \dots, k'$ 个,并对它们进行向量加法,同样可以得到 $n_e = 3, \dots, k'$ 时对应的疑似校验向量。

由上述分析可知,对 $A_{K \times n}$ 依据式(9)进行高斯消元得到矩阵 $F_{n \times n}$,其列向量仅包含码组基中错误码字数满足 $0 \leq n_e \leq 1$ 时的疑似校验向量。当 $2 \leq n_e \leq k'$ 时,需要另外构造疑似校验向量。若将新构造的向量依次排列在 $F_{n \times n}$ 的列向量之后,那么 $F_{n \times n}$ 的列数将变为 $m = n + C_{k'}^2 + \dots + C_{k'}^{n_e}$ 。定义 n_E 为码组基中错误码字数量的上限,即 $0 \leq n_e \leq n_E$,且 $n_E \leq k'$ 。定义 a_i 和 a^i 分别为 $A_{K \times n}$ 第 i 行行向量和第 i 列列向量, a_{ij} 为 $A_{K \times n}$ 第 i 行第 j 列的元素。求解疑似校验向量的容错消元算法,如算法2所示。

算法2 容错高斯消元

输入 $A_{K \times n}, n_E$

输出 $F_{n \times m}$

- 1) 初始化 $F_{n \times m} \leftarrow [I_{n \times n} \ 0_{n \times (m-n)}]$
- 2) for pivot = 1, 2, ..., n do
- 3) $ti \leftarrow 0, tj \leftarrow 0$
- 4) for $i = pivot, \dots, K$ do
- 5) for $j = pivot, \dots, n$ do
- 6) if $a_{ij} = 1$ then
- 7) $ti \leftarrow i, tj \leftarrow j$
- 8) break
- 9) end if
- 10) end for
- 11) if $ti \neq 0$ then break
- 12) end for
- 13) if $ti = 0$ then break
- 14) if $ti \neq pivot$ then
- 15) 交换 a_{pivot} 和 a_{ti}
- 16) end if
- 17) if $tj \neq pivot$ then
- 18) 交换 a^{pivot} 和 a^{tj} , 交换 f^{pivot} 和 f^{tj}
- 19) end if
- 20) for $j = 1, 2, \dots, n$ do
- 21) if $j \neq pivot$ and $a_{pivot,j} = 1$
- 22) $a^j \leftarrow a^j + a^{pivot}, f^j \leftarrow f^j + f^{pivot}$
- 23) end if
- 24) end for

```

25)end for
26)初始化  $sr \leftarrow n + 1$ 
27)初始化  $\{sv_1, sv_2, \dots, sv_{k'}\} \leftarrow \{2, 3, \dots, k', 0\}$ 
28)初始化  $\{sn_1, sn_2, \dots, sn_{k'}\} \leftarrow \{k' - 1, \dots, 0\}$ 
29)for  $n_e = 2, \dots, n_E$  do
30) for  $i = 1, \dots, k' - n_e + 1$  do
31)  for  $j = 0, \dots, sn_i - 1$  do
32)    $f^{sr+j} \leftarrow f^j + f^{sv_i+j}$ 
33)  end for
34)   $sr \leftarrow sr + sn_i$ 
35)   $sv_i \leftarrow sr$ 
36) end for
37) 初始化  $sum \leftarrow 0$ 
38) for  $i = k' - n_e + 1, \dots, 1$  do
39)   $sum \leftarrow sum + sn_i$ 
40)   $sn_i \leftarrow sum - sn_i$ 
41) end for
42) end for

```

算法2包含两部分，步骤1)至步骤25)是对 $A_{K \times n}$ 进行高斯消元的过程，因 $A_{K \times n}$ 行数 $K = n + 10$ ，故此部分计算复杂度为 $O(n^3)$ 。步骤26)至步骤42)是进一步构造疑似校验向量的过程，利用错误码数字量 $n_e = 1$ 时的疑似校验向量构造 $n_e = 2$ 时的疑似校验向量，利用 $n_e = 1$ 和 $n_e = 2$ 时的疑似校验向量构造 $n_e = 3$ 时的疑似校验向量，依次构造错误码数字量满足 $4, \dots, n_e$ 时的疑似校验向量。此过程是一个线性递推的过程，共进行 $m - n$ 次 n 维向量加法操作，

计算复杂度为 $O\left(n \sum_{n_e=2}^{n_E} C_k^{n_e}\right)$ 。综上所述，算法2

的总计算复杂度为 $O\left(n\left(n^2 + \sum_{n_e=2}^{n_E} C_k^{n_e}\right)\right)$ 。

当 $n_E = 1$ 时，算法2本质上等同于高斯消元算法。尽管其计算复杂度仍为 $O(n^3)$ ，与传统高斯消元算法相当，但算法2能够生成容纳单个错误码字的疑似校验向量，从而显著提升了容错能力。

当 $n_E = 2$ 时，算法2相较于 $n_E = 1$ 的情况，额外增加了 C_k^2 次 n 维向量加法操作。由于 k' 是矩阵 $A_{K \times n}$ 的秩，在有误码的条件下 $k \leq k' \leq n$ ，因此 $C_{k'}^2 = \frac{k'(k'-1)}{2} \leq \frac{n(n-1)}{2}$ 。此时，算法2的计算复杂度与进行2次高斯消元的计算复杂度相当。然

而，算法2能够生成将近 $\frac{n+n(n-1)}{2}$ 个疑似校验向量，而进行2次高斯消元最多只能获得 $2n$ 个疑似校验向量，远少于 $n_E = 2$ 时所能生成的疑似校验向量数量。

考虑 $n_E = k'$ 的极端情况，算法2生成的疑似校验向量由两部分组成：第1部分是 $V(2^n)$ 的 $n - k'$ 维子空间的一组基；第2部分是 $V(2^n)$ 的 k' 维子空间。其中，第1部分是 $A_{K \times n}$ 的对偶向量，第2部分是 $A_{K \times n}$ 码组基在 $n_e = 1, 2, \dots, k'$ 时的对偶向量。在这种情况下，算法2枚举了码组基的全部错误可能，必定能够找到码字对偶空间的一组基。然而，此时算法2的总计算复杂度为 $O(n(n^2 + 2^{k'} - k'))$ ，接近 Walsh 变换法^[14-15]，是一个极其庞大的计算量。

因此，为了保证容错性的同时降低计算复杂度，算法2应选取较小的 n_E 值。在本文的仿真实验中，设置 $n_E = 2$ 以实现这一目标。

2.4 疑似校验向量的判定

设疑似校验向量为 \mathbf{h} ，其码重为 w_h 。假设条件 \mathcal{H}_0 表示 \mathbf{h} 是随机向量，假设条件 \mathcal{H}_1 表示 \mathbf{h} 是校验向量。 N_{judge} 表示 \mathbf{h} 校验的码数字量，将码字通过与不通过校验的数量之差 t 作为统计量。当 N_{judge} 充分大且误比特率为 p_e 时，在假设条件 \mathcal{H}_0 和 \mathcal{H}_1 下， t 分别服从式(10)和式(11)所示的正态分布^[15,17]。

$$t \sim \mathcal{N}(0, N_{\text{judge}}) \quad (10)$$

$$t \sim \mathcal{N}\left(N_{\text{judge}}(1 - 2p_e)^{w_h}, N_{\text{judge}}\left(1 - (1 - 2p_e)^{2w_h}\right)\right) \quad (11)$$

设式(10)的均值和方差分别为 μ_0 和 σ_0^2 ，式(11)的均值和方差分别为 μ_1 和 σ_1^2 。根据正态分布的累计分布函数可知，在 \mathcal{H}_0 假设条件下超过 99.99% 的 t 位于区间 $[\mu_0 - 5\sigma_0, \mu_0 + 5\sigma_0]$ 。类似地，在 \mathcal{H}_1 假设条件下超过 99.99% 的 t 位于区间 $[\mu_1 - 5\sigma_1, \mu_1 + 5\sigma_1]$ 。若两区间无重叠，即

$$\mu_0 + 5\sigma_0 < \mu_1 - 5\sigma_1 \quad (12)$$

则能明显区分 t 属于哪个正态分布。将 μ_0 、 σ_0 、 μ_1 和 σ_1 代入式(12)得

$$N_{\text{judge}} > 25 \left(\frac{1 + \sqrt{1 - (1 - 2p_e)^{2w_h}}}{(1 - 2p_e)^{w_h}} \right)^2 \quad (13)$$

由式(12)可知,若要有效利用 t 区分校验向量和随机向量,则 N_{judge} 需要满足式(13)。又因为 \mathbf{Y} 中实际接收码字的数量 N 是有限的,所以

$$N_{\text{judge}} = \min \left[25 \left(\frac{1 + \sqrt{1 - (1 - 2p_e)^{2w_h}}}{(1 - 2p_e)^{w_h}} \right)^2, N, 200 \right] \quad (14)$$

其中,200是为了使 N_{judge} 充分大,能保证 t 服从正态分布。计算最小错误判决门限及其近似值^[8]分别为

$$A_{\text{opt}} = \frac{(\mu_1 \sigma_0^2 - \mu_0 \sigma_1^2) \pm \sigma_0 \sigma_1 \sqrt{(\mu_0 - \mu_1)^2 + 2(\sigma_0^2 - \sigma_1^2) \ln \frac{\sigma_0}{\sigma_1}}}{(\sigma_0^2 - \sigma_1^2)} \quad (15)$$

$$A_{\text{opt}} \approx \frac{\mu_0 \sigma_1 + \mu_1 \sigma_0}{\sigma_0 + \sigma_1} \quad (16)$$

由式(14)可知, $N_{\text{judge}} \leq N$,所以 N_{judge} 不满足式(13)。此时两区间重叠部分较大,虽然 A_{opt} 平均判错概率最小,但是误判概率不容忽视。因此,设置判决下限为

$$A_{\text{opt}} = \max(\Lambda_{\text{opt}}, \mu_0 + 5\sigma_0) \quad (17)$$

从而保持较小的误判概率。当疑似校验向量的统计量 $t \geq A_{\text{opt}}$ 时,即可判定为校验向量。

2.5 校验矩阵重建算法步骤

为了进一步提升校验矩阵重建算法的容错能力,本文还引入了列随机抽取,设码字被抽取的比特数为 s 。经过上述分析,已在 $s = n$ 的条件下,即码字结构完整的条件下,实现了码字筛选、疑似校验向量获取和校验向量判定。当 $s < n$ 时,只需将关于码长的参数 n 替换为 s ,上述方法同样适用,整个校验矩阵的重建算法,如算3所示。

算法3 校验矩阵重建

输入 硬判决序列 \mathbf{y} ,软解调序列 \mathbf{r} ,码长 n 和起点 sp ,误比特率 p_e ,比特抽取数量 s ,列随机抽取次数 iter_{col} ,校验重量上限 w_{max} ,容错上限 $n_E = 2$,权重因子 $\beta = 0.66$,调节因子 $\alpha = 1$

输出 校验矩阵 \mathbf{H}

1)初始化 $K \leftarrow s + 10$, $\mathbf{H} \leftarrow \mathbf{0}$

2)由 \mathbf{y} 和 \mathbf{r} 根据式(1)和式(2)构造 \mathbf{Y} 和 \mathbf{R}

3)由 p_e 和 w_{max} 根据式(14)计算 N_{judge}

4)for $i = 1, 2, \dots, \text{iter}_{\text{col}}$ do

5) 随机抽取 \mathbf{Y} 和 \mathbf{R} 构造 \mathbf{Y}_s 和 \mathbf{R}_s

6) 由 \mathbf{Y}_s 、 \mathbf{R}_s 和 K 根据算法1对 \mathbf{Y}_s 进行排序

7) 由 \mathbf{Y}_s 的前 K 行构造 $\mathbf{A}_{K \times s}$

8) 根据算法2对 $\mathbf{A}_{K \times s}$ 消元,获取 $\mathbf{F}_{s \times m}$

9) 删除 $\mathbf{F}_{s \times m}$ 中校验重量大于 w_{max} 的疑似校验向量

10) 随机从 \mathbf{Y}_s 中选择 N_{judge} 个码字

11) 计算疑似校验向量的统计量 t

12) 由式(17)计算疑似校验向量的判决下限 A_{opt}

13) 将 $t \geq A_{\text{opt}}$ 的疑似校验向量还原为校验向量

14) 将不能被 \mathbf{H} 表示出的校验向量加入 \mathbf{H}

15) 利用新增校验向量和调节因子 α 更新 \mathbf{R}

16)end for

在整个重建过程中,算法3共执行 iter_{col} 次码字筛选和容错高斯消元。其中,高斯消元的计算复杂度是 $O(s^3)$,码字筛选的计算复杂度是 $O(nN)$ 。当 s 较大时,码字筛选的计算量相较于高斯消元可忽略不计,因此其总计算复杂度为 $O(\text{iter}_{\text{col}}s^3)$ 。本文方法优先选取高可靠度码字重建校验矩阵,通过新增校验向量进一步提高码字可靠度,从而加速了迭代收敛,而且不需要行随机抽取。因此,本文算法的计算复杂度显著低于传统算法^[8-11]。

3 仿真实验及性能分析

本节基于MATLAB 2022a平台开展实验研究。由于RS码和汉明码是BCH码的特例,故实验选取部分BCH码以及IEEE802.11n协议和802.11e标准中的LDPC码作为研究对象。实验首先验证码字筛选与对数似然比更新的有效性,随后通过与传统高斯消元算法对比,评估容错高斯消元算法的容错性能。最后系统分析码长、码率、码字数量、列抽取次数以及调节因子和权重因子对本文算法重建性能的影响,并在重建率和计算复杂度方面与现有算法进行对比。

3.1 码字筛选有效性验证

在不同的信噪比(SNR, signal to noise)条件下,分别随机生成140 000个BCH(63,30)码字和IEEE802.11n协议下的LDPC(648,324)码字,统计正确、错误以及全部比特对数似然比绝对值的均值,如图4所示。

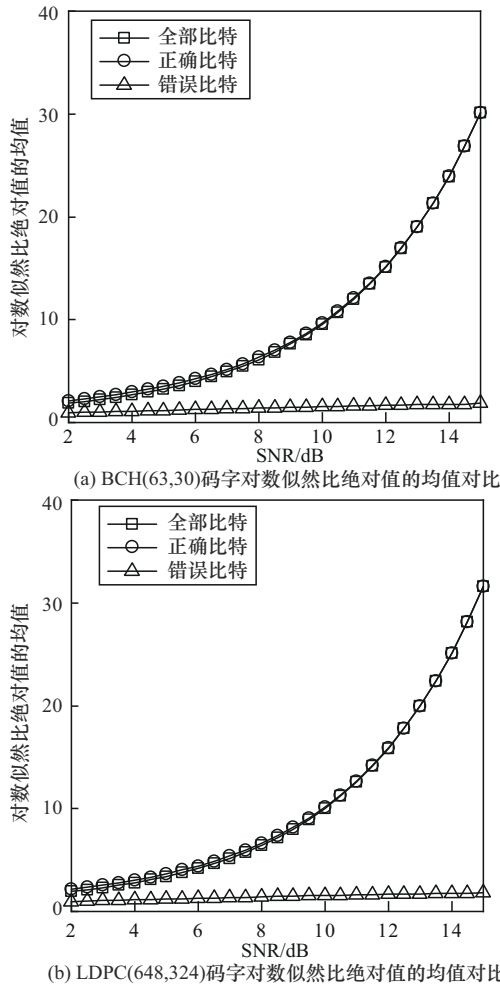


图4 对数似然比绝对值的均值对比

由图4可以看出，在不同的SNR条件下，正确比特与全部比特对数似然比绝对值的均值相差不大，而正确比特与错误比特对数似然比绝对值的均值存在明显差异。对数似然比绝对值的均值大小反映了比特可靠度的高低，因此可以用来区分正确比特和错误比特。

在不同SNR条件下，随机生成140 000个BCH(63, 30)码字和IEEE802.11n协议中的LDPC(648,324)码字，根据式(1)和式(2)，将它们分别排列成矩阵并按算法1进行排序，进行10 000次蒙特卡罗仿真。统计码字的整体误比特率，以及排序前后矩阵前K行码字的误比特率，如图5所示。

由图5可知，排序前矩阵前K行码字的误比特率与整体误比特率一致，而排序后矩阵前K行码字的误比特率则低于整体误比特率。随着SNR的降低，排序后矩阵前K行码字的误比特率与整体误比特率的差异逐渐减小，虽然筛选效果逐步下降，但

是依旧优于未筛选效果，因此该算法能够从整体码字中有效筛选出部分高质量码组。

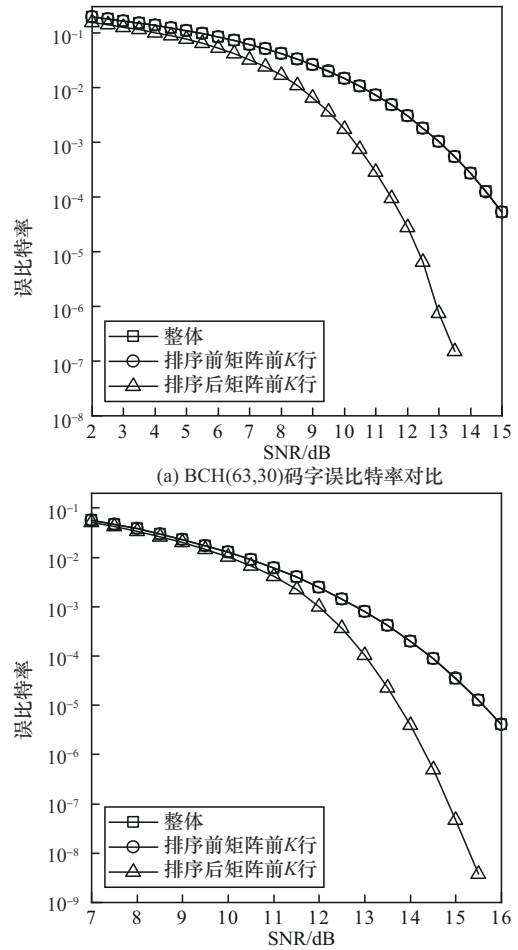


图5 误比特率对比

在误比特率为0.000 051 4的条件下，随机生成3 000个IEEE802.11n协议中的LDPC(648, 324)码字，根据式(1)和式(2)，将其排列成码字矩阵并按算法1进行排序。标记排序前后码字矩阵的误比特率分布，如图6所示。

对比图6(a)和图6(b)可知，排序前码字矩阵的误比特率分布均匀，排序后码字矩阵的误比特率分布大多集中在矩阵下方，取矩阵上部几行码字时，其质量较排序前有显著提升。这是因为排序后，矩阵上部码字可靠度高，且出错概率低，而下部码字可靠度低，更易出错。因此，该算法能有效筛选出高质量码字用于校验矩阵重建。

3.2 对数似然比更新有效性验证

在误比特率为0.05时，随机生成10 000个BCH(63,30)码字。设列抽取次数为200，比特抽取

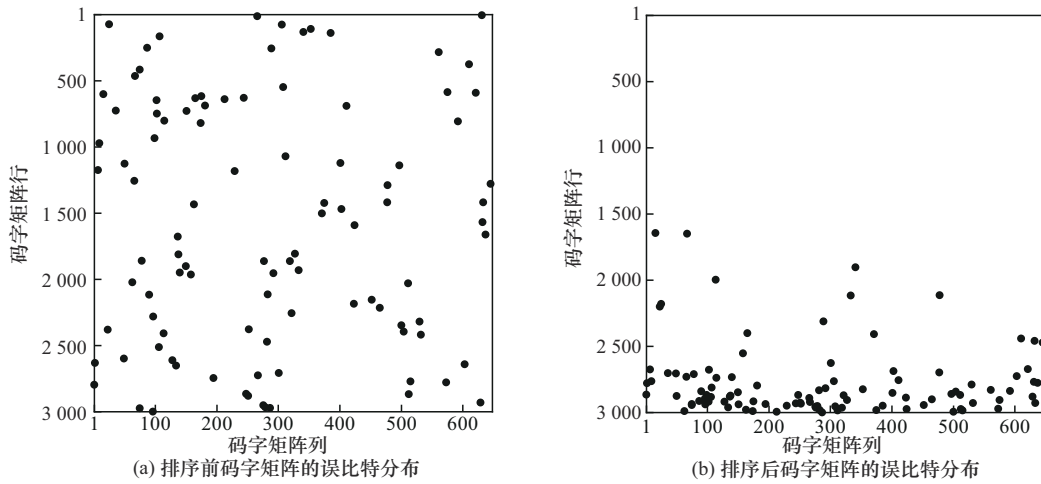


图6 误比特分布

数量为30, 校验向量最大重量为18, 容错上限为2。消元矩阵 $A_{40 \times 30}$ 的误比特率随列抽取次数的变化情况, 如图7(a)所示。在误比特率为0.006时, 随机生成3 000个IEEE802.16e标准中定义的LDPC(576, 288)码字。设列抽取次数为3 000, 比特抽取数量为288, 校验向量最大重量为7, 容错上限为2。消元矩阵 $A_{298 \times 288}$ 的误比特率随列抽取次数的变化情况, 如图7(b)所示。

由图7可知, 随着列抽取次数增加, 消元矩阵的误比特率先整体下降, 最后趋于稳定。这是因为列抽取次数增多, 重建的校验向量就越多, 更多来自校验向量的可靠度信息被更新到比特可靠度中。当比特可靠度足够高时, 消元矩阵的误比特率可能降为0, 如图7(a)所示。在校验向量完全重建后, 比特可靠度不变, 消元矩阵的误比特率趋于平稳, 如图7(b)所示。

3.3 容错性能验证

在不同误比特率下, 随机生成BCH(63,30)码组和IEEE802.11n协议中的LDPC(648,324)码组, 其中BCH(63,30)码组包含73个码字, LDPC(648,324)码组包含658个码字。针对每个码组, 根据式(1)构造码字矩阵, 并通过直接高斯消元、列高斯消元以及容错上限为1和2的容错高斯消元对其进行变换, 分别得出各码组的疑似校验向量。随后, 统计不同消元算法所得疑似校验向量中真实且线性无关的校验向量数量, 结果如图8所示。

由图8可知, 在相同误比特率下, 容错高斯消元的容错能力强于直接高斯消元和列高斯消元, 并且会随着容错上限的提高而增强。原因在于容错高斯消元通过求解部分错误状态下的疑似校验向量, 增大了被破坏校验关系的恢复概率。而且容错上限越高, 枚举的错误状态越多, 容错能力越强。

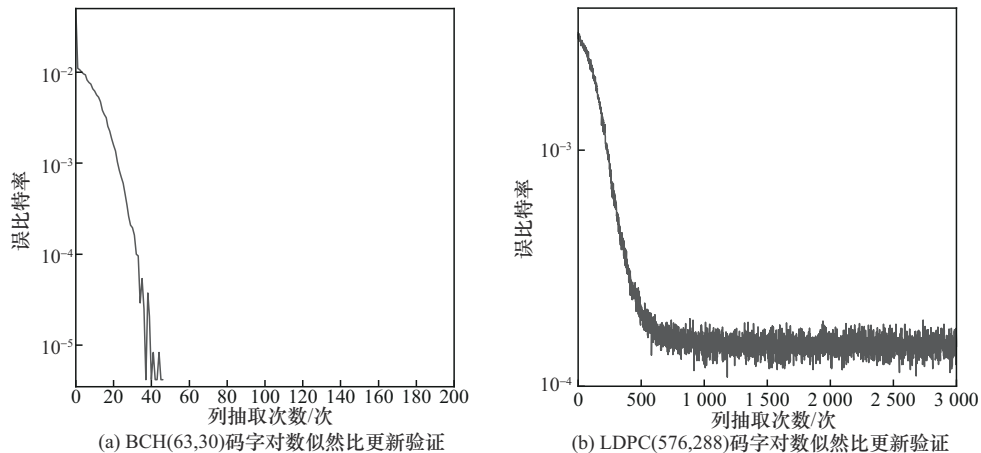


图7 对数似然比更新验证

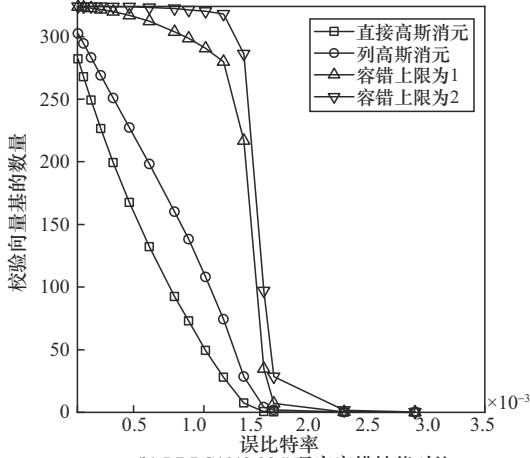
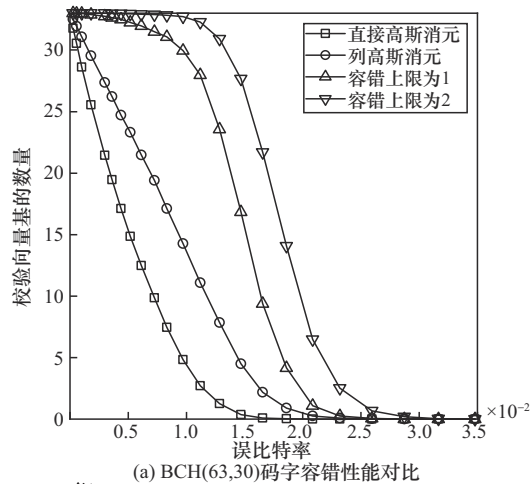


图8 容错性能对比

3.4 码长对重建的影响

在不同误比特率下，分别随机生成 10 000 个 BCH(15,7)、BCH(31,16)和 BCH(63,30)码字。设列抽取次数为 200，比特抽取数量为各自信息长度，校验向量最大汉明重量分别为 4、8 和 18，容错上限为 2，统计本文算法校验矩阵的重建率，如图 9 所示。

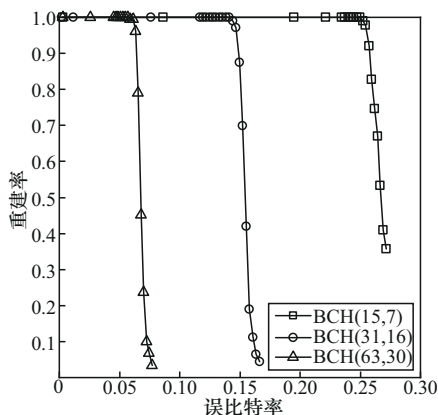


图9 码长对校验矩阵重建的影响

由图 9 可知，在误比特率相同的条件下，随着码长的增大，校验矩阵的重建率降低。这主要基于两方面原因：一方面是校验向量的汉明重量会随码长的增大而增加，这既提高了校验关系被误比特破坏的概率，也使得因不满足式(13)而漏判的情况增多；另一方面是比特抽取数量随码长的增大而增多，导致抽到错误比特的概率上升，进而使求解出的校验向量数量减少。

3.5 码率对重建的影响

在不同误比特率下，分别随机生成 3 000 个 IEEE802.11n 协议中码率为 $\frac{1}{2}$ 、 $\frac{2}{3}$ 和 $\frac{3}{4}$ 且码长为 648 的 LDPC 码字。设列抽取次数为 3 000，比特抽取数量为各码率对应的信息长度，校验向量最大重量依次为 8、11 和 15，文献[11]迭代消元次数为 20，本文算法容错上限为 2。统计文献[11]与本文算法的校验矩阵重建率，结果如图 10 所示。

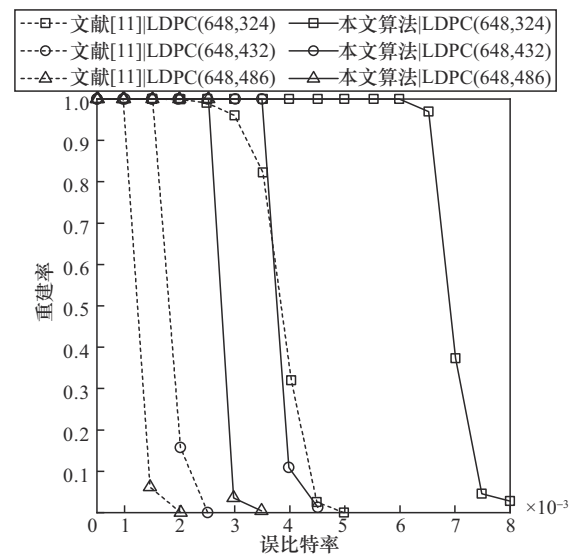


图10 码率对校验矩阵重建的影响

由图 10 可知，在误比特率相同的条件下，随着码率的增大，校验矩阵的重建率降低。同样是因为码率增大导致校验向量的汉明重量增大和比特抽取数量增大，从而导致校验矩阵的重建率降低。仿真结果表明，在码率分别为 $\frac{1}{2}$ 、 $\frac{2}{3}$ 和 $\frac{3}{4}$ 的情况下，文献[11]在误比特率分别达到 0.005、0.002 5 和 0.002 时已无法正常工作。而本文算法在误比特率为 0.006 5、0.003 5 和 0.002 5 时，仍能稳定保持 96% 以上的校验矩阵重建率。因此，与文献[11]相比，本文算法具有更强的容错性和健壮性。

3.6 码字数量对重建的影响

在不同误比特率下, 分别随机生成 1 500、3 000、6 000 和 10 000 个 IEEE802.11n 协议中 LDPC(648, 324)码字。设列抽取次数为 3 000, 比特抽取数量为 324, 校验向量最大重量为 8, 文献[11]迭代消元次数为 20, 本文算法容错上限为 2。统计文献[11]与本文算法的校验矩阵重建率, 结果如图 11 所示。

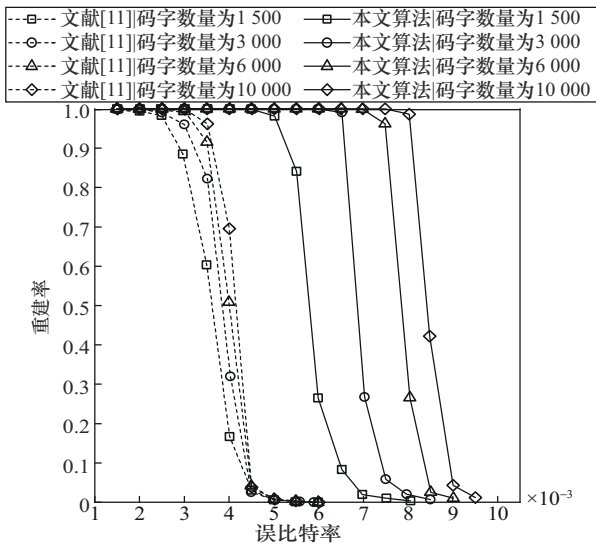


图 11 码字数量对校验矩阵重建的影响

由图 11 可知, 在误比特率相同的情况下, 码字数量越多, 校验矩阵的重建率越高。这是由于码字数量增多时, 针对某一校验关系, 未被破坏的码字也更多, 从而求解出该校验关系的概率更大。仿真结果显示, 在 LDPC(648,324)码字数量为 10 000 的条件下, 当本文算法的误比特率达到 0.008 时, 校验矩阵的重建率已超过 98%。相比之下, 文献[11]在误比特率仅为 0.005 5 时便已失效。这一结果主要归因于两方面: 一是本文算法具有更强的容错能力和更高的鲁棒性; 二是在每次消元过程中, 本文算法会从码组中筛选出最可靠的码字进行消元处理, 而文献[11]则采用随机选取的方式。因此, 随着码字数量的增加, 本文算法的性能提升效果显著优于文献[11]。

3.7 列抽取次数对重建的影响

在误比特率分别为 0.003 5 与 0.006 时, 随机生成 3 000 个 IEEE802.16e 标准中定义的 LDPC(576,288)码字。设定比特抽取数量为 288, 校验向量最大重量为 7, 文献[11]的迭代消元次数为 20, 而本文算法的容错上限为 2。统计不同列抽取次数下本文算法与文献[11]的校验向量的重建数量, 如图 12 所示。

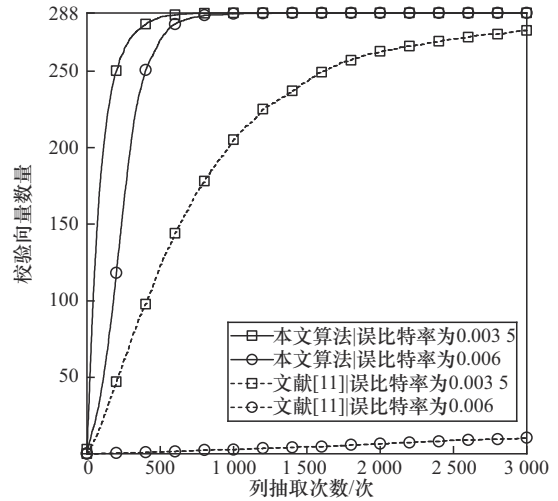


图 12 列抽取次数对校验矩阵重建的影响

由图 12 可知, 在误比特率相同的情况下, 列抽取次数越多, 校验向量的重建数量就越多。这是因为: 一方面, 多次迭代提高了校验关系式被抽取和求解的概率; 另一方面, 已求出的校验关系式会提升未求出关系式被求解的概率, 每次求出一个校验关系式, 都会增强码字比特的可靠性, 进而增加剩余关系式被求解的可能性。仿真结果表明, 在列抽取次数为 1 000 次的情况下, 当误比特率为 0.003 5 和 0.006 时, 本文算法的校验矩阵重建率均超过 99%。文献[11]在相同列抽取次数下表现较差, 当误比特率为 0.003 5 时, 其校验矩阵重建率不足 75%。当误比特率上升至 0.006 时, 重建率更是降至不到 1%。此外, 在误比特率为 0.003 5 时, 本文算法重建出 250 个校验向量需要约 200 次抽取, 而文献[11]需要约 1 600 次抽取。由此可见, 本文算法不仅具有更强的容错能力和更高的鲁棒性, 还能够在更低的计算复杂度下高效完成校验矩阵的重建任务。

3.8 权重因子和调节因子对重建的影响

在误比特率为 0.065 时, 随机生成 10 000 个 BCH(63,30)码字。设列抽取次数为 200, 比特抽取数量为 30, 校验向量最大重量为 18, 容错上限为 2。在误比特率为 0.008 4 时, 随机生成 3 000 个 IEEE 802.16e 标准中定义的 LDPC(576,288)码字。设列抽取次数为 3 000, 比特抽取数量为 288, 校验向量最大重量为 7, 容错上限为 2。首先, 将调节因子 α 的取值范围设定为 0~10, 间隔为 1, 同时将权重因子 β 的值设为 0.66, 分别统计两组码字的校验矩阵的重建率, 如图 13(a)所示。然后, 将权重因子 β 的取值

范围设定为0~1, 间隔为0.1, 并在0.6~0.7将间隔设为0.02, 同时将调节因子 α 的值设为1, 分别统计两组码字的校验矩阵的重建率, 如图13(b)所示。

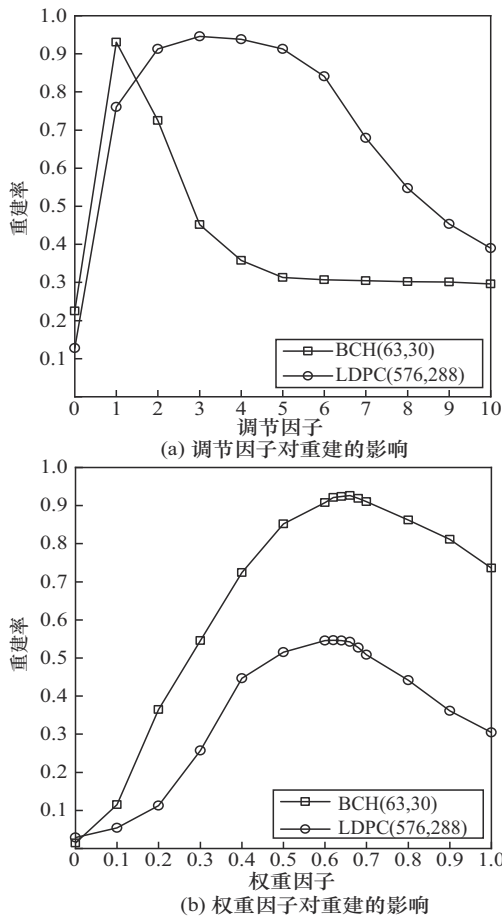


图13 调节因子和权重因子对重建的影响

由图13(a)可知, 在相同误比特率下, 随着调节因子的增大, 校验矩阵的重建率呈现先升后降的趋势。这是因为调节因子反映了校验向量提供的信息, 其取值需要权衡, 若调节因子过小(如等于零), 则无法有效反映比特的可靠性, 而若调节因子过大, 则会过度依赖校验向量信息, 忽略比特本身的可靠性。因此, 比特的可靠性应综合考虑其对数似然比的绝对值以及是否通过校验的结果来评估, 而非单纯依赖调节因子。此外, 从图中还可以看出, 不同线性分组码的最佳调节因子存在一定差异。但由于本文不针对具体线性分组码的特例展开研究, 故统一将调节因子设定为1, 这一取值基本能够反映校验向量提供的信息。

由图13(b)可知, 在相同误比特率下, 随着权重因子的增大, 校验矩阵的重建率同样呈现先升后降

的趋势。这是由于码字的可靠度需综合考虑其内部比特可靠度的均值和最小值特征, 否则难以全面反映码字的整体可靠性, 这与2.2节中的码字可靠度分析一致。此外, 从图中还可以看出, 不同线性分组码的最佳权重因子差异较小, 通常在0.6~0.7取得最优值。因此, 本文统一将权重因子设定为0.66, 这一取值能够在均值和最小值特征之间实现较好的平衡。

3.9 实际计算时间对比

首先在误比特率为0.003 5的条件下, 分别随机生成 $K = k + 10$ 个 LDPC(576,288)、LDPC(648,324)、LDPC(648,432)和LDPC(648,486)码字。然后分别构建码字矩阵并从中随机抽取 k 列构造 $A_{K \times k}$, 其中 k 为各自码字的信息位长度。在基准频率为3.70 GHz的 Intel i5-12600KF CPU 和 MatLab 2022a 的仿真环境下, 利用算法2(容错高斯消元)和传统高斯消元算法对 $A_{K \times k}$ 进行消元, 各算法实际计算时间对比如表1所示。

| 算法 | LDPC(576, 288)/s | LDPC(648, 324)/s | LDPC(648, 432)/s | LDPC(648, 486)/s |
|--------|------------------|------------------|------------------|------------------|
| 容错高斯消元 | 0.034 | 0.046 | 0.11 | 0.16 |
| 传统高斯消元 | 0.015 | 0.019 | 0.05 | 0.08 |

由表1可知, 算法2的实际计算复杂度与2.3节的理论分析一致, 其计算时间约为传统高斯消元算法的2倍。这是因为算法2额外引入了 C_k^2 次 n 维向量模2加操作, 其中 $C_k^2 \leq \frac{n(n-1)}{2}$ 。因此, 算法2额外增加的计算量最多为 $\frac{n^2(n-1)}{2}$ 次模2加操作。该结果进一步验证了当容错上限 $n_E = 2$ 时, 算法2的计算复杂度与传统高斯消元算法计算复杂度具有相同的数量级, 即计算复杂度为 $O(n^3)$ 。

4 结束语

为提高误码条件下线性分组码校验矩阵的重建率, 本文提出了一种基于容错高斯消元的校验矩阵重建算法。该算法首先通过随机抽取部分码字比特构建残缺码组并筛选出高可靠度码字。然后利用容错高斯消元算法求解校验向量。最后, 通过已知校验向量更新码字的可靠度, 反复迭代直至完成校验

矩阵的重建。与传统重建算法相比,本文算法显著提升了重建所用码字的质量,充分利用了码组中未破坏的校验关系,增强了疑似校验向量的求解能力并加速了迭代收敛。仿真结果表明,在相同条件下,本文算法重建率高于传统重建算法,具备良好的抗误码性能和实际应用价值。

参考文献:

- [1] 张永光,楼才义.信道编码及其识别分析[M].北京:电子工业出版社,2010.
ZHANG Y G, LOU C Y. Channel coding and recognition analysis[M]. Beijing: Publishing House of Electronics Industry, 2010.
- [2] 吴昭军,张立民,钟兆根,等.基于平均余弦符合度下的本原BCH码盲识别[J].通信学报,2020,41(1):15-24.
WU Z J, ZHANG L M, ZHONG Z G, et al. Blind recognition of primitive BCH code based on average cosine conformity[J]. Journal on Communications, 2020, 41(1): 15-24.
- [3] 吴昭军,张立民,钟兆根,等.一种软判决下的RS码识别算法[J].电子与信息学报,2020,42(9):2150-2157.
WU Z J, ZHANG L M, ZHONG Z G, et al. Blind recognition of RS codes based on soft decision[J]. Journal of Electronics & Information Technology, 2020, 42(9): 2150-2157.
- [4] 尤红雨,王焱,王翔,等.基于码重奇偶性的扩展BCH码盲识别[J].系统工程与电子技术,2024,46(5):1783-1791.
YOU H Y, WANG Y, WANG X, et al. Blind identification of extended BCH codes based on parity of code weight[J]. Systems Engineering and Electronics, 2024, 46(5): 1783-1791.
- [5] 尤红雨,王焱,王翔,等.误码条件下TPC码盲识别[J].信号处理,2023,39(7):1251-1261.
YOU H Y, WANG Y, WANG X, et al. Blind recognition of TPC code under error code condition[J]. Journal of Signal Processing, 2023, 39(7): 1251-1261.
- [6] 张天骐,王雪怡,方竹,等.基于软判决下的校验关系的循环码参数盲识别[J].信号处理,2023,39(6):1120-1130.
ZHANG T Q, WANG X Y, FANG Z, et al. Soft-decision based blind identification of cyclic code parameters with parity-check relation[J]. Journal of Signal Processing, 2023, 39(6): 1120-1130.
- [7] 张永光,楼才义,王焱.一种线性分组码编码参数的盲识别方法[P].中国:CN102201882B,2014.
ZHANG Y G, LOU C Y, WANG T. A blind parameters identification method for linear block code[P]. China: CN102201882B, 2014.
- [8] 吴昭军,张立民,钟兆根,等.高误码率下LDPC稀疏校验矩阵重建[J].通信学报,2021,42(3):1-10.
WU Z J, ZHANG L M, ZHONG Z G, et al. Reconstruction of sparse check matrix for LDPC at high bit error rate[J]. Journal on Communications, 2021, 42(3): 1-10.
- [9] 刘恒燕,张立民,闫文君,等.基于随机抽取的稀疏校验矩阵重建算法[J].系统工程与电子技术,2023,45(4):1215-1221.
LIU H Y, ZHANG L M, YAN W J, et al. Reconstruction algorithm of sparse check matrix based on random extraction[J]. Systems Engineering and Electronics, 2023, 45(4): 1215-1221.
- [10] 王忠勇,李正豪,巩克现,等.高误码率下基于随机抽取的LDPC码校验矩阵重建[J].通信学报,2023,44(3):128-137.
WANG Z Y, LI Z H, GONG K X, et al. Reconstruction of LDPC code check matrix based on random extraction at high bit error rate[J]. Journal on Communications, 2023, 44(3): 128-137.
- [11] 张天骐,李春运,吴仙越,等.基于改进型LBP译码的LDPC码稀疏校验矩阵重建[J].通信学报,2024,45(5):70-79.
ZHANG T Q, LI C Y, WU X Y, et al. Reconstruction of LDPC code sparse check matrix based on modified LBP decoding[J]. Journal on Communications, 2024, 45(5): 70-79.
- [12] 包昕,周磊珂,何可,等.误码条件下的LDPC码盲识别算法[J].西安交通大学学报,2015,49(12):53-58.
BAO X, ZHOU L K, HE K, et al. A recognition algorithm for LDPC codes of blind in a noisy environment[J]. Journal of Xi'an Jiaotong University, 2015, 49(12): 53-58.
- [13] 刘杰,张立民,占超.基于矩阵分析的线性分组码盲识别[J].系统工程与电子技术,2017,39(2):404-409.
LIU J, ZHANG L M, ZHAN C. Blind recognition of linear block codes based on matrix analysis[J]. Systems Engineering and Electronics, 2017, 39(2): 404-409.
- [14] 游凌,朱中梁. Walsh函数在解二元域方程组上的应用[J].信号处理,2000,16(S1):27-30,20.
YOU L, ZHU Z L. The application of Walsh function in resolving of F(2) equations[J]. Journal of Signal Processing, 2000, 16(S1): 27-30, 20.
- [15] 杨晓炜,甘露.基于Walsh-Hadamard变换的线性分组码参数盲估计算法[J].电子与信息学报,2012,34(7):1642-1646.
YANG X W, GAN L. Blind estimation algorithm of the linear block codes parameters based on WHT[J]. Journal of Electronics & Information Technology, 2012, 34(7): 1642-1646.
- [16] 王焱,满欣,尤红雨,等.矩阵变换特征与码序列联合学习的卷积码识别方法[J].国防科技大学学报,2023,45(5):38-47.
WANG Y, MAN X, YOU H Y, et al. Convolutional codes recognition method based on joint learning of matrix transformation features and code sequences[J]. Journal of National University of Defense Technology, 2023, 45(5): 38-47.
- [17] CHABOT C. Recognition of a code in a noisy environment[C]//Proceedings of the 2007 IEEE International Symposium on Information Theory. Piscataway: IEEE Press, 2007: 2211-2215.

[作者简介]



王忠勇(1965-),男,江西遂川人,博士,郑州大学教授、博士生导师,主要研究方向为通信信号处理、嵌入式系统等。

张洪(1997-),男,河南商丘人,郑州大学硕士生,主要研究方向为信道编码分析。

巩克现(1976-),男,山东泰安人,博士,郑州大学教授、博士生导师,主要研究方向为无线通信信号分析与处理、信道编码、无线接入、目标监测及电子对抗等。

王玮(1974-),女,黑龙江密山人,博士,郑州大学副教授、硕士生导师,主要研究方向为无线通信和信号处理等。

张玲(1976-),女,安徽颖上人,博士,郑州大学高级工程师,主要研究方向为卫星信号处理、网络安全等。

朱俊昊(1998-),男,河南信阳人,郑州大学硕士生,主要研究方向为带宽无线通信、数字信号处理。