

一种基于增强型奇偶校验码改进的极化码 EPC-MS-SCL译码算法

张降龙¹,袁建国²,游 薇²,余林峰²,黄 胜¹

(1. 重庆邮电大学通信与信息工程学院,重庆 400065;2. 重庆邮电大学光电工程学院,重庆 400065)

摘 要: 针对基于路径分裂策略辅助极化码串行抵消列表(Path Splitting Selecting strategy based on Search Set under the Successive Cancellation List, PSS-SS-SCL)译码算法性能的不足,提出了一种利用增强型奇偶校验码改进 PSS-SS-SCL 译码算法的 EPC-MS-SCL(Enhanced Parity Check and Monte Carlo Segment aided Successive Cancellation List)译码算法. 该算法在极化码编码阶段对信息序列做分段处理,在每段末尾添加增强型奇偶校验码,译码器仅在译码搜索集内元素时进行路径分裂,其余元素直接执行硬判决译码,并在译码完一段序列后立即对该段进行校验,仅保留通过校验的路径,从而减少了错误路径对正确路径的竞争,使正确路径保留到译码结束的概率增加,改善了译码性能,同时减少了译码列表数,使得译码复杂度更低. 仿真结果表明,与 PSS-SS-SCL 译码算法相比较,所提出算法能在一定程度上改善其性能增益且具有更低的译码复杂度.

关键词: 极化码; PSS-SS-SCL 译码; 搜索集; 增强型奇偶校验码; 剪枝

基金项目: 国家自然科学基金(No.61971079);重庆市研究生科研创新项目(No.CYS21301)

中图分类号: TN911

文献标识码: A

文章编号: 0372-2112(2023)09-2391-07

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20211678

An Improved EPC-MS-SCL Decoding Algorithm of Polar Codes Based on the Enhanced Parity Check Code

ZHANG Xiang-long¹, YUAN Jian-guo², YOU Wei², YU Lin-feng², HUANG Sheng¹

(1. School of Communication and Information Engineering,

Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

2. School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: In order to improve the performance of the path splitting selecting strategy based on search set under the successive cancellation list (PSS-SS-SCL) decoding algorithm, an improved decoding algorithm based on the enhanced parity check code and Monte Carlo segment aiding successive cancellation list (EPC-MS-SCL) is proposed. The algorithm divides the information sequence into some segments in the encoding stage of the polar code, and adds an enhanced parity check code at the end of each segment. The decoder performs the path splitting only when decoding the elements within the search set (SS), and directly does the hard decision for the other unfrozen bits. The decoder will verify the segment immediately after decoding a sequence, the verified path can only be retained. Thus the competition from the wrong paths to the correct path can be reduced to increase the probability of keeping the correct path until the end of the decoding and the performance of the decoding can be improved. Meanwhile, the decoding complexity is also reduced because of decreasing the number of the decoding lists. Simulation results show that compared with the PSS-SS-SCL decoding algorithm, the proposed decoding algorithm can improve the coding gain to a certain degree and reduce the decoding complexity too.

Key words: polar code; PSS-SS-SCL decoding, search set; enhanced parity check code; pruning

Foundation Item(s): National Natural Science Foundation of China (No.61971079); Graduate Student Scientific Research Innovation Projects in Chongqing City (No.CYS21301)

1 引言

极化码由 Arkan 提出^[1],是第一类能够被证明达到二进制离散无记忆对称信道(binary input discrete memoryless symmetric channel)的信道容量的码字,是新一代移动通信系统的重要技术. 极化码已经成功入选5G标准,成为5G增强移动宽带场景下控制信道的编码方案,是当前信道编码领域的研究热点. 除信道容量可达特性外,极化码的突出优势是在串行抵消(Successive Cancellation, SC)译码算法下不存在误码平台(Error Floor),然而在码长有限长的情况下,由于信道极化不完全,实际SC译码算法性能并不理想^[2,3]. 于是文献[4]提出了能够保留多条译码路径的串行抵消列表(Successive Cancellation List, SCL)译码算法,文献[5]采用与循环冗余校验(Cyclic Redundancy Check, CRC)码级联的CRC-Polar码,提出循环冗余校验码辅助串行抵消列表(CRC Aid SCL, CA-SCL)译码算法,提高了SCL译码算法的译码性能,同时也是目前主流的极化码译码方案. 尽管CA-SCL译码算法具有不错的译码性能,但该算法具有较高的译码复杂度 $O(LN \log_2 N)$,其中 L 为译码器列表数大小, N 为极化码码字长度. 为了降低CA-SCL译码算法的译码复杂度,研究者们提出了一系列改进方案. 文献[6]和文献[7]提出了Segment-CRC译码算法,通过分段检验的方式减少译码失败后译码器后续不必要的译码操作来降低译码复杂度. 由于极化码不同比特的可靠性之间存在显著的差异,不需要对全部信息比特进行分裂,基于此,文献[8]提出一种减少路径分裂次数的方案:利用极化信道偏序关系构造关键集,译码器仅在译码关键集中的元素时进行路径分裂,减少译码过程中路径的分裂次数,从而降低CA-SCL译码复杂度. 类似地,文献[9]提出基于路径分裂策略的PSS-SS-SCL(Path Splitting Selecting strategy based on Search Set under the Successive Cancellation List)译码算法,该算法译码器仅在译码搜索集中元素时进行路径分裂,在几乎和CA-SCL译码算法性能相同的情况下,获得更低的译码复杂度. 文献[9]通过减少译码时路径分裂次数从而实现了译码复杂度的降低,然而上述两种方案未考虑如何提高分裂比特位置处的译码成功率,因此相较于CA-SCL译码算法并未取得译码性能上的提升. 针对该问题,本文通过分析SCL译码算法译码失败的原因,并利用增强型奇偶校验(Enhanced Parity Check, EPC)码改进文献[9]中的PSS-SS-SCL译码算法,从而提出一种既保留了PSS-SS-SCL译码算法低分裂次数特点,又取得更好的译码性能的EPC-MS-SCL(Enhanced Parity Check Monte Carlo Segment Successive Cancellation List)译码算法.

2 极化码译码

2.1 极化码SC译码

极化码的SC译码算法是序贯译码算法,从首个比特开始逐个译码信源比特,直到译完最后一个比特为止.

设 W 是二进制输入无记忆对称信道,输入符号集合为 $X \in \{0, 1\}$,输出符号集合为 Y ,信道转移概率记为 $W(y|x)$,其中 $x \in X = \{0, 1\}$, $y \in Y$. 信道 W 复用 N 次,以传输码字序列 $\mathbf{x}_1^N = \mathbf{u}_1^N \mathbf{G}_N$,其中 \mathbf{u}_1^N 表示输入至极化码编码器的信源序列, \mathbf{G}_N 表示极化码生成矩阵. 接收序列记为 \mathbf{y}_1^N ,使用对数似然比(Logarithm Likelihood Ratio, LLR)作为译码过程中的可靠性度量. 第 i 个极化信道 $W_N^{(i)}(\mathbf{y}_1^N, \mathbf{u}_1^{i-1}|u_i)$ 的对数似然比 $L_N^{(i)}$ 定义如下:

$$L_N^{(i)}(\mathbf{y}_1^N, \mathbf{u}_1^{i-1}|u_i) = \ln \frac{w_N^{(i)}(\mathbf{y}_1^N, \mathbf{u}_1^{i-1}|u_i=0)}{w_N^{(i)}(\mathbf{y}_1^N, \mathbf{u}_1^{i-1}|u_i=1)} \quad (1)$$

使用LLR作为译码度量时,比特判决准则如下:

$$u_i = \begin{cases} 0, & i \in A^c \text{ 或 } L_N^{(i)} \geq 0 \\ 1, & i \in A \text{ 且 } L_N^{(i)} < 0 \end{cases} \quad (2)$$

其中 A^c 表示极化码冻结比特集合,通常将其值置为0, A 表示极化码非冻结比特集合.

2.2 极化码CA-SCL译码

CA-SCL译码器在译码每一个非冻结比特时进行路径分裂,分别保存译码可能结果0和1,并按式(3)所示更新每条路径的路径度量(Path Metric, PM)^[10]值. 第 l 条译码路径在译码第 i 个比特后的路径度量值 PM_l^i 的计算如式(3):

$$PM_l^i = PM_l^{i-1} + \ln(1 + e^{-(1-2\hat{u}_{l,i})L_N^i(\mathbf{y}_1^N, \mathbf{u}_1^{i-1})}) \quad (3)$$

在实际使用中,可使用式(4)来简化表示:

$$PM_l^i = \begin{cases} PM_l^{i-1}, & \text{if } \hat{u}_{l,i} = \frac{1}{2}(1 - \text{sgn}(L_N^i(l))) \\ PM_l^{i-1} + |L_N^i(l)|, & \text{else} \end{cases} \quad (4)$$

其中 $\hat{u}_{l,i}$ 表示第 l 条译码路径中第 i 个比特的译码估计值. 当分裂后的路径数超过最大译码列表数 L 时,仅保留 L 条PM值较小的路径. 在译完最后一个比特后,从通过CRC校验的路径中选择PM值最小的路径作为CA-SCL译码器的输出码字.

2.3 极化码PSS-SS-SCL译码

极化码可以看作由多个子极化码块组成,如图1所示,图中白色叶子节点表示冻结比特,黑色叶子节点表示信息比特,灰色节点表示既包含冻结比特又包含信息比特的子块,其中仅包含信息比特的子块 $\{A, B, C, D, E\}$ 被称为码率1(R1)节点^[11]. PSS-SS-SCL译码算法选择每个R1节点的首位元素 $\{u_4, u_8, u_{10}, u_{11}, u_{13}\}$ 作为搜索集中的元素. PSS-SS-SCL译码算法在译码搜索集元素时采用SCL译码得到每个元素的译码估计,

而对搜索集以外的元素则采用SC译码得到每个元素的译码估计.

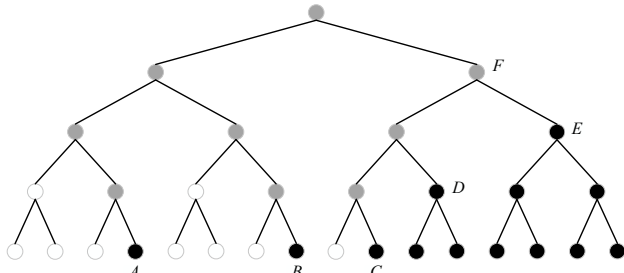


图1 码长为16的极化码二叉树示意图

3 SCL译码失败原因分析

SCL译码失败的主要原因在于译码过程中丢失掉正确路径,本文对译码过程中正确路径丢失的原因分析如下.

假设发送信息序列为全0码字,当前译码第 K 个非冻结比特,在译完第 $K-1$ 个非冻结比特时,存活的路径数量为 m ,并且第 m 条路径中保留着正确译码结果,左侧 PM_i 为每条路径在译码第 K 个非冻结比特前的路径度量值,右侧 PM_{i1}, PM_{i2} 为每条译码路径分裂后的路径度量值,如图2所示.

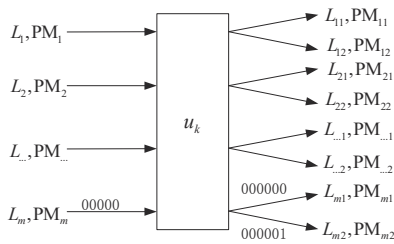


图2 SCL译码第 K 个非冻结比特

假设 L_m, L_{m1} 为正确路径,下面分两种情况对译码结果分析.

情况1,当 u_k 硬判决能正确译码时:

此时 $PM_{m1} < PM_{m2}$,当且仅当路径 L_{m1}, L_{m2} 都没有被保留下来,才会导致SCL译码失败.因此,对于可靠的位置,采用SC译码可避免路径 L_{m1}, L_{m2} 都不被保留下来的情况,从而改善该位置译码成功率.

情况2,当 u_k 硬判决不能正确译码时:

此时 $PM_{m1} > PM_{m2}$,当且仅当路径 L_{m1}, L_{m2} 都被保留下来才会使得SCL在该位置成功译码.因此,对于某些容易受噪声影响导致硬判决出错的位置,在此位置译码前减少错误路径数量,降低错误路径对正确路径的竞争,可以提高该位置保留正确路径的能力,从而改善该位置译码成功率.

由于文献[12]在仿真实验中验证了搜索集包含了

约99%的受信道噪声导致硬判决出错的信道,结合上述情况1与情况2的分析,本文提出相应的改进方案,即在搜索集中错误率较高的元素前使用EPC码进行校验剪枝,减少一部分错误路径以提高这些信道保留正确路径的能力,从而获得更好的译码性能.

4 增强型奇偶校验码

由于传统的奇偶校验码只能校验奇数个错误,无法校验偶数个错误,其校验能力不强;CRC码的校验能力较强却存在校验过程运算复杂度较高的缺点.针对上述两种校验码的不足,本文提出一种EPC码,该校验码使用多个校验位,某一位校验比特分别为信息序列中某些位的奇偶校验,以两位EPC码为例:

设待校验序列为 x_1^k ,校验序列 m_1^2 表示使用两位EPC码,此时校验位 m_i 的值由式(5)~(7)得到:

$$m_i = \sum_{j=1}^{n_i} x_{1+(j-1)d}, \quad d = 2^{i-1} \quad (5)$$

$$n_1 = k \quad (6)$$

$$n_i = \begin{cases} n_{i-1}/2, & n_{i-1} \text{为偶数} \\ (n_{i-1}+1)/2, & n_{i-1} \text{为奇数} \end{cases} \quad (7)$$

对上式作解释:第一位校验位的值由序列 x_1^k 所有元素进行传统奇偶校验编码所得,第二位校验位的值由第一位校验码所校验序列中奇数位进行奇偶校验编码所得,对于第 n 位校验位的值,由第 $n-1$ 位校验序列中的奇数位进行奇偶校验编码所得,如图3所示.

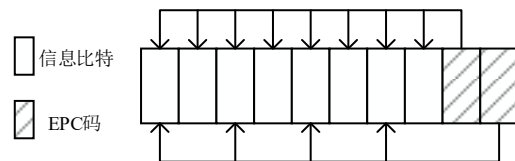


图3 待校验序列比特数为8,两位EPC码

对EPC码检错能力分析,以两位EPC码为例:

(1)当序列 x_1^k 中出现奇数个错误,可通过 m_1 检测出来.

(2)当序列 x_1^k 中出现偶数个错误,此时分为序列 x_1^k 中奇数位和偶数位各自出现奇数个错误以及各自出现偶数个错误两种情况,对于前一种情况,可通过 m_2 检测出来.

因此,两位EPC码可以检测出所有的奇数个错误以及一半的偶数个错误(本文假设奇数位和偶数位出现错误的概率是一样的),此时检错能力为:

$$\frac{1}{2} + \frac{1}{2} \times \frac{1}{2} = \frac{3}{4} \quad (8)$$

对于 n 位EPC码,经推导其检错能力 $D_n = \frac{1}{2} +$

$\sum_{i=1}^{n-1} (\frac{1}{4})^i, n \geq 2$ (n 为校验码的位数). 由此可知, 随着校验码位数的增加, 其校验能力也相应增加, 然而过长的校验码会增加信息的冗余度. 基于此, 本文采用 1~2 位 EPC 码用于改进 PSS-SS-SCL 译码算法.

5 一种改进的 EPC-MS-SCL 译码算法

5.1 编码

设极化码码长为 N , 传输的信息比特序列为 \mathbf{u}_1^k , 其中码率 $R = \frac{K}{N}$, 额外添加 EPC 码的数量为 m , 额外添加 CRC 码的数量为 r .

在编码时首先通过高斯近似算法^[13]选择 $k+m+r$ 个较好的信道用于传输非冻结比特集合 A , 其余 $N-k-m-r$ 个信道用于传输冻结比特集合 A^c , 再根据 A 与 A^c 搜索出极化码中 R1 节点并构造搜索集. 接着采用 Monte Carlo 仿真实验来统计搜索集中每个元素的错误率, 具体步骤为: 发送端发送全 0 码字, 在接收端采用 PSS-SS-SCL 译码算法进行译码, 在译码搜索集元素 S_i 时, 若在译码前存在正确路径且译码 S_i 后也存在正确路径, 则记该次译码在 S_i 处成功; 若译码 S_i 后不存在正确路径则记该次译码在 S_i 处失败, 通过足够多的重复实验统计得到每一个搜索集元素的错误率.

在通过 Monte Carlo 仿真实验统计出搜索集元素的错误率之后, 选择其中前 m 个错误率最大的元素, 并将位于这些元素前的 1~2 个信道用于传输 EPC 码, 每个 EPC 码的值由式 (5)~(7) 计算得到, 其中第 i 个 EPC 码所校验序列为第 $i-1$ 个 EPC 码至第 i 个 EPC 码中间的信息比特集合, 如图 4 所示.

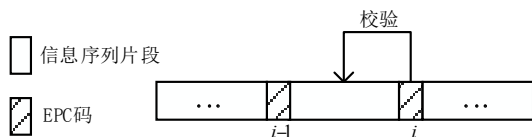


图 4 第 i 个 EPC 码所校验的信息序列

将信息序列 \mathbf{u}_1^k 通过 m 位 EPC 码编码后得到序列 \mathbf{u}_1^{k+m} , 接着在序列 \mathbf{u}_1^{k+m} 的末尾添加 r 位 CRC 码用于校验前面所有比特从而得到序列 \mathbf{u}_1^{k+m+r} , 最后将序列 \mathbf{u}_1^{k+m+r} 作为极化码的非冻结比特集合 A 进行极化码编码得到极化码码字 \mathbf{x}_1^N . 具体步骤如算法 1 所示. 添加 EPC 码和 CRC 码后的序列 \mathbf{u}_1^{k+m+r} 如图 5 所示, 其中取 $k=128, m=8, r=8$.

分析算法 1 可知, 本文提出的改进方案相较于 PSS-

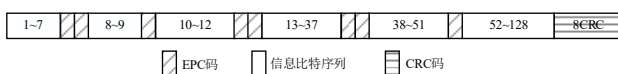


图 5 添加 EPC 码与 CRC 码后的序列 $\mathbf{u}_1^{128+8+8}$

SS-SCL 译码算法在编码部分增加了统计搜索集元素的错误率以及进行 EPC 码编码这两个步骤. 其中 EPC 码编码时仅需遍历 m (m 为添加 EPC 码的总数) 个搜索集中错误率较高的元素, 并且计算 EPC 码的值仅涉及复杂度低且易于硬件实现的模二加法运算, 因此对原信息序列分段并添加 EPC 码这一步骤增加的复杂度几乎可以忽略. 其中统计搜索集元素错误率的复杂度与执行 Monte Carlo 仿真实验的次数成正比, 执行的次数越多, 统计得到搜索集元素的错误率越准确, 但相应的复杂度就越高, 而在实际使用时可以提前将不同码长及码率情况下的搜索集元素的错误率进行统计并存储下来, 从而将这一步骤的时间复杂度降为 $O(1)$.

算法 1 EPC-MS-SCL 编码步骤

输入: 码长 N , 信息比特序列 \mathbf{u}_1^k , EPC 码数目 m , CRC 码数目 r

输出: 极化码码字 \mathbf{x}_1^N

- 1: 通过高斯近似算法选择 $k+m+r$ 个较好的信道用于传输非冻结比特集合 A , 其余 $N-k-m-r$ 个信道用于传输冻结比特集合 A^c ;
- 2: 通过集合 A^c, A 构造搜索集;
- 3: 通过 Monte Carlo 仿真实验, 得到搜索集中每个元素的错误率, 选择其中前 m 个错误率最大的元素, 并按信道顺序进行升序排列;
- 4: 遍历 m 个元素;
- 5: 首位元素通过判断位于该位置前面信息比特个数, 若前面没有信息比特, 则跳过该元素; 若信息比特个数小于 5, 则第一段信息序列末尾放置 1 个校验比特; 其他情况则放置 2 个校验比特;
- 6: 非首位元素则比较与前一个元素之差, 若等于 1, 跳过当前元素; 若小于 5, 在该段信息序列末尾放置 1 个校验比特; 其他情况则在该段信息序列末尾放置 2 个校验比特;
- 7: 上述遍历直到 m 个校验比特放置完毕, 此时得到 EPC 码编码后序列 \mathbf{u}_1^{k+m} ;
- 8: 在序列 \mathbf{u}_1^{k+m} 末尾添加 r 位 CRC 码, 得到序列 \mathbf{u}_1^{k+m+r} ;
- 9: 将序列 \mathbf{u}_1^{k+m+r} 作为极化码的非冻结比特集合 A 进行极化码编码, 得到极化码码字 \mathbf{x}_1^N ;

5.2 译码

EPC-MS-SCL 译码过程在 PSS-SS-SCL 译码的基础上增加了 EPC 码校验步骤, 由于 EPC 码在校验时仅涉及模二加法运算, 因此 EPC 码校验步骤增加的复杂度几乎可以忽略. 具体译码步骤如算法 2 所示. 整个编译码流程如图 6 所示.

6 仿真分析

在本节中分析不同算法误块率以及各自的译码复杂度. 分析 CA-SCL 译码算法可知, 译码复杂度主要与三个步骤有关:

(1) 用于译码每个比特的路径数, 路径数越小, 译码过程中使用的译码列表数越少, 译码复杂度越低.

(2) 译码过程中不分裂比特的数量, 不分裂比特越多, 译码复杂度越低.

算法 2 EPC-MS-SCL 译码步骤

输入: 码长 N , 冻结比特集合 A^c , 接收信号集合 y , 搜索集, EPC 码位置集合 P
 输出: 信息序列估计 \hat{u}^N

- 1: FOR $i = 1:N$
- 2: IF $i \in A^c$
- 3: $\hat{u}_i = 0$;
- 4: ELSE
- 5: IF $i \in$ 搜索集
- 6: 每条路径执行 SCL 译码得到 \hat{u}_i ;
- 7: ELSE
- 8: 每条路径执行 SC 译码得到 \hat{u}_i ; END IF;
- 9: END IF;
- 10: IF $i \in P$
- 11: 每条路径对当前段信息序列进行 EPC 码校验, 仅保留通过校验的路径;
- 12: IF 所有路径均未通过校验
- 13: 宣布本次译码失败, 停止后续译码步骤; END IF;
- 14: END IF;
- 15: END FOR;
- 16: IF 存在通过 CRC 校验路径
- 17: 选择其中 PM 值最小的路径作为译码输出序列 \hat{u}^N ;
- 18: ELSE
- 19: 宣布本次译码失败;
- 20: END IF;

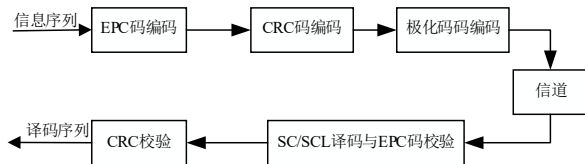


图6 整体编译码流程图

(3) 译码过程中排序的次数, 译码过程中排序次数越少, 译码复杂度越低。

因此本文通过统计平均一次译码列表总数, 平均一次译码分裂比特总数以及平均一次译码排序总数来反映各算法译码复杂度的高低。其中, 平均一次译码列表总数 L_{sum} 为译码每个比特使用的路径总和, 其值越大实际译码列表总数越大, 译码复杂度越高。 L_{sum} 计算如式(9)所示。

$$L_{sum} = \sum_{i=1}^N L_i \quad (9)$$

其中 L_i 表示译码第 i 个比特时使用的路径数。

为了公平地比较每个算法的误块率与译码复杂度, 本文在仿真实验中对每个算法设置相同的码长 N , 信息比特数 K , 其中码率 $R = \frac{K}{N}$, 并且各算法额外加入的校验码数量均相同, 图7~10显示了各个算法在不同极化码码长 N , 码率 R 下的误块率仿真结果。其中, CA-

SCL 译码算法与 PSS-SS-SCL 译码算法采用在信息序列末尾添加 16 位 CRC 校验码; Segment-CRC 译码算法将信息序列均匀分为 4 段且每段末尾添加 4 位 CRC 校验码; EPC-MS-SCL 译码算法采用 8 位 EPC 码和 8 位 CRC 码。各算法均采用 BPSK 调制, 信道选用加性高斯白噪声信道。图 11~13 显示了各算法在码长 $N=256$, 码率 $R = \frac{1}{2}$ 时, 不同列表大小下的译码复杂度仿真结果。

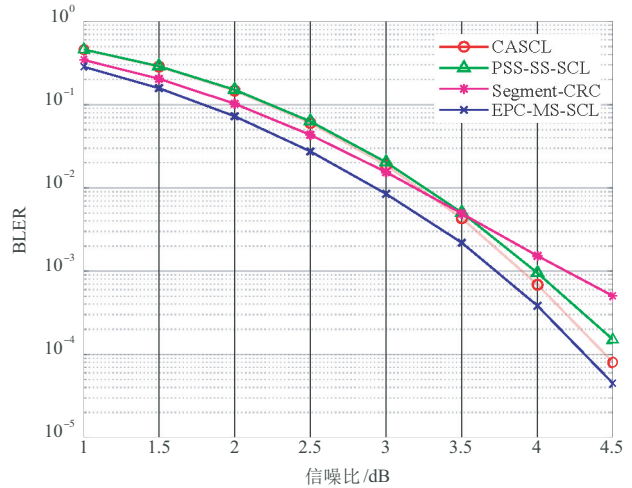


图7 $N=128, R = \frac{1}{3}$ 时各算法误块率仿真图

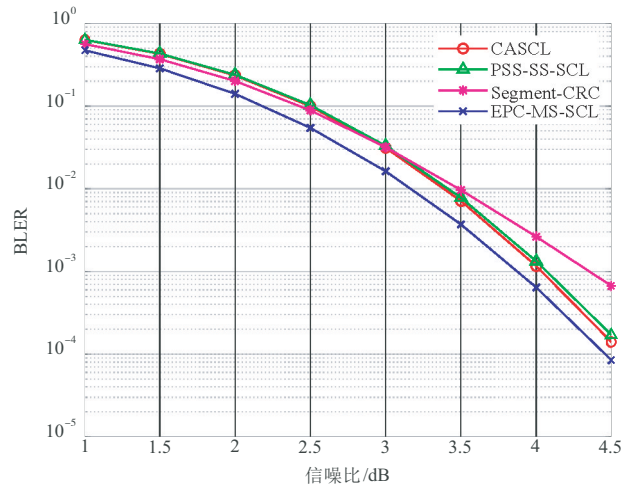
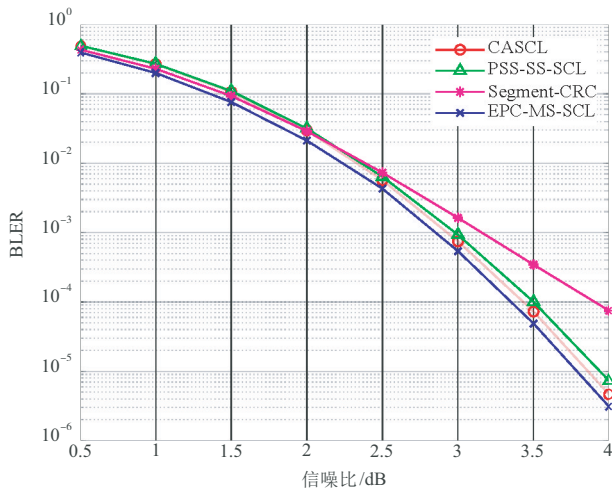


图8 $N=128, R = \frac{1}{2}$ 时各算法误块率仿真图

由图7~10可看出, 本文所提出的 EPC-MS-SCL 译码算法较其他三种译码算法拥有更低的误块率, 其中, EPC-MS-SCL 译码算法在码长 $N=128$, 码率 $R = \frac{1}{3}$, $BLER = 10^{-3}$ 时较 CA-SCL 译码算法与 PSS-SS-SCL 译码算法分别约有 0.18 dB 与 0.2 dB 的增益提升; 在码长 $N=256$, 码率 $R = \frac{1}{2}$, $BLER = 10^{-4}$ 时较 CA-SCL 译码算法与 PSS-SS-SCL 译码算法分别约有 0.08 dB 与 0.15 dB 的增益提升。

图9 $N=256, R=\frac{1}{3}$ 时各算法误块率仿真图

由图 11 可知,本文所提出的 EPC-MS-SCL 译码算法较传统 CA-SCL 以及 PSS-SS-SCL 译码算法的平均一次译码列表总数更低,仅略高于 Segment-CRC 译码算法. 在 $L=32$ 时, EPC-MS-SCL 译码算法较传统 CA-SCL 以及 PSS-SS-SCL 译码算法的一次译码列表总数减少了约 5%. 由图 12 可知 EPC-MS-SCL 译码算法与 PSS-SS-SCL 译码算法分裂次数几乎相同,且相较于 CA-SCL 译码算法与 Segment-CRC 译码算法减少了约 70% 的分裂

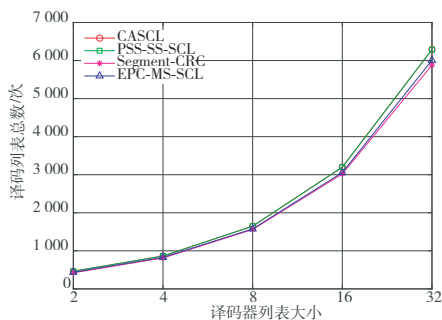


图 11 不同译码算法的平均译码列表总数

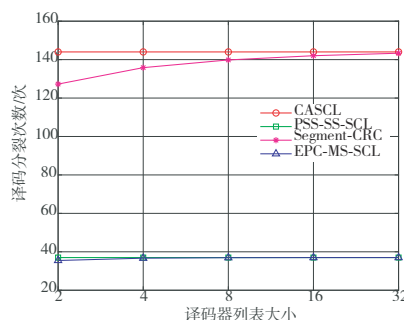


图 12 不同译码算法的平均分裂次数

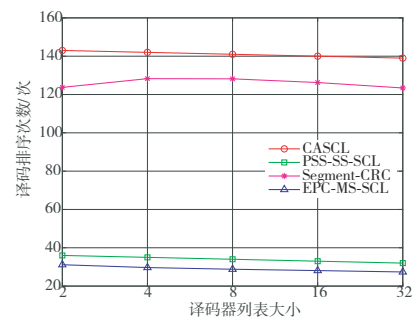
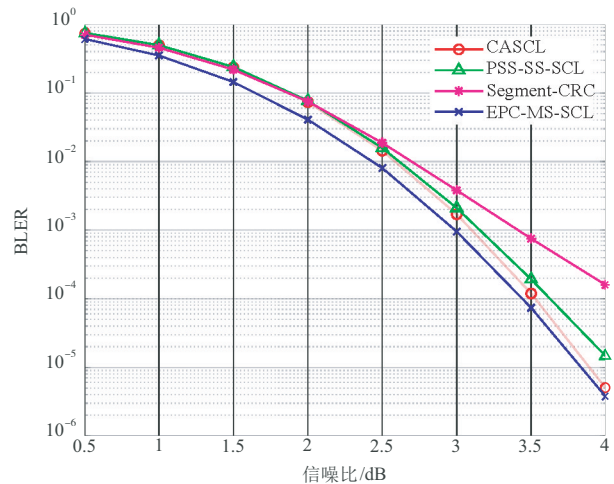


图 13 不同译码算法的平均排序次数

综上所述,本文所提出的 EPC-MS-SCL 算法相较于 PSS-SS-SCL 译码算法不仅保留了其低分裂次数和低排序次数的优点,降低了译码复杂度,并且明显改善了 PSS-SS-SCL 译码算法的译码性能.

7 结束语

本文通过分析 SCL 译码失败的原因以及 PSS-SS-SCL 译码算法的不足,结合 PSS-SS-SCL 译码算法搜索集的构造,提出了一种利用增强型奇偶校验码改进的极化码 EPC-MS-SCL 译码算法. 该算法在极化码编码阶段对信息序列做分段处理,在每段末尾添加增强型奇偶校验码,译码器仅在译码搜索集内元素时进行路

图 10 $N=256, R=\frac{1}{2}$ 时各算法误块率仿真图

次数. 其中,在 $L=2$ 时出现分裂次数稍低于 PSS-SS-SCL 的原因是分段校验具有提前终止功能,在某段译码结束后,若无路径通过校验,则停止后续译码步骤. 由图 13 可知 EPC-MS-SCL 译码算法的平均排序次数相较于传统 CA-SCL 译码算法降低了约 79%,且低于 Segment-CRC 以及 PSS-SS-SCL 译码算法,具有最少的排序次数. 因而本文所提出的 EPC-MS-SCL 译码算法相较于其他三种译码算法具有更低的译码复杂度.

径分裂,其余元素直接执行硬判决译码,并在译码完一段序列后立即对该段进行校验,仅保留通过校验的路径. 这样,不仅减少了错误路径对正确路径的竞争,增加正确路径保留到译码结束的概率,改善了译码性能,而且减少了译码列表数,降低译码复杂度. 仿真验证了该算法相较于传统的 CA-SCL, PSS-SS-SCL 以及 Segment-CRC 译码算法在改善译码性能的同时减少了译码复杂度,为极化码译码方向的研究提供了一个新思路.

参考文献

- [1] ARIKAN E. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input

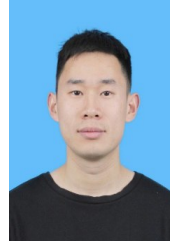
memoryless channels[J]. IEEE Transactions on Information Theory, 2009, 55(7): 3051-3073.

- [2] 李小文, 李文彬. 一种简化的极化码串行消除列表译码算法[J]. 重庆邮电大学学报(自然科学版), 2019, 31(4): 488-494.
LI X W, LI W B. A simplified successive cancellation list decoding algorithm for polar codes[J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2019, 31(4): 488-494. (in Chinese)
- [3] 刘伟, 段红光. 极化码自适应连续消除列表比特翻转译码算法[J]. 重庆邮电大学学报(自然科学版), 2021, 33(1): 87-93.
LIU W, DUAN H G. Adaptive successive cancellation list bit-flip decoding of polar codes[J]. Journal of Chongqing University of Posts and Telecommunications(Natural Science Edition), 2021, 33(1): 87-93. (in Chinese)
- [4] TAL I, VARDY A. List decoding of polar codes[J]. IEEE Transactions on Information Theory, 2015, 61(5): 2213-2226.
- [5] NIU K, CHEN K. CRC-aided decoding of polar codes[J]. IEEE Communications Letters, 2012, 16(10): 1668-1671.
- [6] ZHOU H Y, ZHANG C, SONG W Q, et al. Segmented CRC-aided SC list polar decoding[C]//2016 83rd Vehicular Technology Conference. Nanjing: IEEE, 2016: 1-5.
- [7] ZHOU H Y, LIANG X, LI L P, et al. Segmented successive cancellation list polar decoding with tailored CRC[J]. Journal of Signal Processing Systems, 2019, 91(8):923-935.
- [8] 于永润. 极化码编译码技术研究[D]. 南京:东南大学, 2019.
YU Y R. Investigations on the Construction and Decoding of Polar Codes[D]. Nanjing: Southeast University, 2019. (in Chinese)
- [9] GAO C, LIU R K, DAI B, et al. Path splitting selecting strategy-aided successive cancellation list algorithm for polar codes[J]. IEEE Communications Letters, 2019, 23(3): 422-425.
- [10] BALATSOUKAS-STIMMING A, PARIZI M B, BURG A. LLR-based successive cancellation list decoding of polar codes[J]. IEEE Transactions on Signal Processing, 2015, 63(19): 5165-5179.
- [11] ALAMDAR-YAZDI A, KSCHISCHANG F R. A simplified successive-cancellation decoder for polar codes[J]. IEEE Communications Letters, 2011, 15(12): 1378-1380.
- [12] ZHANG Z Y, QIN K J, ZHANG L, et al. Progressive bit-flipping decoding of polar codes over layered critical sets [C]//2017 IEEE Global Communications Conference. Sin-

gapore: IEEE, 2017: 1-6.

- [13] TRIFONOV P. Efficient design and decoding of polar codes[J]. IEEE Transactions on Communications, 2012, 60(11): 3221-3227.

作者简介



张降龙 男, 1998年2月出生, 四川广安人. 重庆邮电大学信息与通信工程学院硕士研究生, 主要研究方向为通信系统中极化码编译码技术.

E-mail: zzzxianglong@163.com



袁建国(通讯作者) 男, 1968年生于重庆市长寿区. 现为重庆邮电大学教授, 硕士生导师, 主要研究方向为通信系统中极化码技术与LDPC编译码技术.

E-mail: yyyjg@126.com



游薇 女, 1997年11月出生, 四川广安人. 重庆邮电大学光电工程学院硕士研究生, 主要研究方向为通信系统中极化码编译码技术.

E-mail: 1411701075@qq.com



余林峰 男, 1998年9月出生, 四川内江人. 重庆邮电大学光电工程学院硕士研究生, 主要研究方向为通信系统中极化码编译码技术.

E-mail: 1767353097@qq.com



黄胜 男, 1974年生于湖北省英山县, 现为重庆邮电大学教授, 博士生导师, 主要研究方向为通信系统中极化码技术与LDPC编译码技术.

E-mail: huangs@cqupt.edu.cn