

基于分段循环冗余校验的极化码自适应连续取消列表译码算法

王琼 罗亚洁* 李思舫

(重庆邮电大学通信与信息工程学院 重庆 400065)

摘要: 针对极化码连续取消列表(SCL)译码算法为获取较好性能而采用较多的保留路径数, 导致译码复杂度较高的缺点, 自适应SCL译码算法虽然在高信噪比下降低了一定的计算量, 却带来了较高的译码延时。根据极化码的顺序译码结构, 该文提出了一种分段循环冗余校验(CRC)与自适应选择保留路径数量相结合的SCL译码算法。仿真结果表明, 与传统CRC辅助SCL译码算法、自适应SCL译码算法相比, 该算法在码率 $R=0.5$ 时, 低信噪比下(-1 dB)复杂度降低了约21.6%, 在高信噪比下(3 dB)复杂度降低了约64%, 同时获得较好的译码性能。

关键词: 极化码; 自适应译码; 连续取消列表; 分段循环冗余校验

中图分类号: TN929.5

文献标识码: A

文章编号: 1009-5896(2019)07-1572-07

DOI: 10.11999/JEIT180716

Polar Adaptive Successive Cancellation List Decoding Based on Segmentation Cyclic Redundancy Check

WANG Qiong LUO Yajie LI Sifang

(Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: Considering the problem that using a large number of reserved paths causes higher complexity in order to obtain better performance for polar code Successive Cancellation List (SCL) decoding, the adaptive SCL decoding algorithm at a high Signal to Noise Ratio (SNR) reduces a certain amount of calculations, however, brings a higher decoding delay. According to the order of polar code decoding, an SCL decoding algorithm combining segmentation Cyclic Redundancy Check (CRC) with adaptively selecting the number of reserved paths is proposed. The simulation results show that compared with the traditional CRC-assisted SCL decoding algorithm and adaptive-SCL algorithm, when the code rate is $R=0.5$, the complexity under low SNR (-1 dB) is reduced by about 21.6%, and the complexity at high SNR (3 dB) is reduced by about 64%, at the same time, better decoding performance is obtained.

Key words: Polar codes; Adaptive decoding; Successive Cancellation List (SCL); Segmentation Cyclic Redundancy Check (CRC)

1 引言

极化码(polar codes)是由Arikan^[1]提出的一种新型信道编码方案。它基于信道极化(channel polarization)进行设计^[2], 是一种在二进制离散无记忆信道(Binary-Discrete Memoryless Channel, B-DMC)下, 通过严格数学方法证明能够达到信道容量的编码方案。译码采用逐次消除(Successive Cancellation, SC)译码算法, 因其编译码复杂度均

为 $O(N\log_2 N)$, 成为广泛使用的解码算法之一^[3]。当码长无限长时, 极化码性能可达理论最优。对短码和长码情况, 极化码展现出比低密度奇偶校验码(Low Density Parity Check code, LDPC)以及Turbo码更优的综合性能, 因此极化码被选作第五代移动通信(5th-Generate communication, 5G) eMBB场景下控制信道的编码方案。

SC算法在路径扩展中只保留一条扩展路径, 容易导致正确路径丢失, 因此文献^[4]中使用了连续取消列表(Successive Cancellation List, SCL)算法, 以广度优先替代深度优先。SCL算法最大保留 L 条候选路径, 使正确的路径有更大的可能进行下一层路径扩展。SCL算法及其衍生算法成为了一套功能强大并在不断改进中的算法^[5]。在译码保留路

收稿日期: 2018-07-17; 改回日期: 2019-01-14; 网络出版: 2019-01-25

*通信作者: 罗亚洁 664136256@qq.com

基金项目: 国家科技重大专项基金(2018ZX03001026-002)

Foundation Item: The National Science and Technology Major Project of China (2018ZX03001026-002)

径中选择最终译码序列时, 引入循环冗余校验码(Cyclic Redundancy Check, CRC)与极化码级联进一步提升译码性能, 即CRC辅助SCL译码算法(CRC-Aided SCL, CA-SCL)^[6]。CA-SCL虽然获得了较好的性能但是其复杂度增大为 $O(LM\log_2 N)$, 其中 L 为列表大小。分段CRC辅助SCL译码算法(Segmented CRC-Adided SCL, SCA-SCL)基于CA-SCL提出, 该算法降低了一定的复杂度, 但性能只能与CA-SCL算法设置列表大小为 L 时近似, 而没有提升^[7]。在硬件实现过程中, 为了降低占用面积和功耗, 文献^[5]提出分区SCL译码算法(Partitioned SCL, PSCL), 与传统的SCL解码相比, PSCL中减少的存储器以误码性能降低为代价。同时为了减少延迟并提高吞吐量, 简化的SCL译码算法^[8](Simplified SCL, SSCL)和快速简化的SCL译码算法^[9](Fast Simplified SCL, Fast-SSCL), 它们依赖于位模式的识别来修剪SC解码树并减少所需估算的位数, 但纠错性能均有下降。

为了获得更好的纠错性能, 往往会选择保留更多的路径, 但随之而来的是更高的计算复杂度和时延。基于CA-SCL的自适应SCL译码算法(ADaptive SCL, AD-SCL)^[10], 根据译码序列的CRC校验结果判断是否增大 L 值重新译码。在高信噪比下, 该算法降低了整体译码复杂度同时获得较好的译码性能。在低信噪比下, 该算法重新译码次数较多, 时延较高。因此本文提出一种基于分段CRC的自适应SCL译码算法(Segmented-CRC ADaptive SCL, SCAD-SCL), 通过对发送码字添加多段CRC, 根据极化码的顺序译码结构, 可在前一部分译码后进行CRC校验, 判断是否增大 L 值重新译码, 而无需将接收信号全部译码后再判断, 避免了无用的计算。该算法在保证译码性能的前提下, 降低了平均复杂度和时延。

2 极化码基本原理

极化码的构造是根据链式法则对比特信道进行联合(channel combining)和分裂(channel splitting)。信道联合相当于对单个独立的比特信道 \mathbf{W} 进行了 M 次复用, 并将其作为一个矢量信道, 再经过虚拟信道分裂可以得到 N 个子信道, 并且有 $M=N$ 从而在独立的比特信道之间引入相关性。根据信道极化原理^[1]可知, 当码长无限长时, 信道分裂后的子信道将呈现两极化现象, 部分子信道的信道容量趋于“1”, 称为无噪信道, 而另一部分子信道的信道容量趋于“0”, 称为纯噪声信道, 其中无噪信道占有所有子信道的比例为 $I(\mathbf{W})$ 。当传输数据时, 只要将信息比特放在无噪信道上, 而纯噪信道发送冻结比特, 也称固定比特(一般为“0”

bit), 理论上就能够达到香农限。实际上编码都是在有限码长条件下进行的, 因此我们只在有限极化子信道中选择可靠度较高的子信道传输信息比特, 可靠度较低子信道传输冻结比特。

2.1 极化码构造

极化码构造可分为3步: (1)极化子信道选择; (2)比特混合; (3)极化码编码。其中最关键的就是对子信道的选择。因为不同信道适用的子信道排序算法不同, 置信序列也就不一样。当前常用子信道可靠度排序方法有: Arikan在文献^[1]中提出的适用于二进制擦除信道的巴氏参数法; Mori等人^[11]从LDPC码中引入密度进化法(Density Evolution, DE); 文献^[12]在DE基础上提出的适用于高斯信道的高斯近似法(Density Evolution-Gaussian Approximation, DE-GA); Schurch在文献^[13]中揭示极化码子信道间的偏用通序关系, 华为提出了利用极化权重(Polarization Weight, PW), 并引入 β 参数扩展, 来计算AWGN信道下子信道可靠度的算法^[14]。

得到置信序列后, 就可以将信息比特与冻结比特按照极化码原理进行混合, 即选择可靠度高的位置放置信息比特, 其余位置置0, 得到混合序列。

极化码编码过程与线性分组码相似, 是基于通用2阶极化核 $\mathbf{F} = [1 \ 0; 1 \ 1]$, 经过 n 次kronecker乘积得到生成矩阵 $\mathbf{G} = \mathbf{F}^{\otimes n}$, 再用混合序列 u 与生成矩阵相乘得到编码码字: $d = u * \mathbf{G}$ 。

2.2 极化码译码

SC译码算法, 首先定义 $P(N, K, \mathbf{A}, \mathbf{A}^c)$ 是码长为 N 、信息比特长度为 K 的极化码, 其中 \mathbf{A} 为信息集合, \mathbf{A}^c 为冻结集。译码过程中先判断第 i 个比特是否属于 \mathbf{A}^c , 若属于则直接判为0, 不属于则判决有两种可能的取值: $\hat{u}_i = 0$ 和 $\hat{u}_i = 1$ 。分别计算它们在信道下传输的转移概率 $\mathbf{W}_N^{(i)} = (y_1^N, \hat{u}_1^{i-1} | \hat{u}_i)$, 然后计算其对数似然比

$$\mathbf{L}_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) = \ln \frac{\mathbf{W}_N^{(i)} = (y_1^N, \hat{u}_1^{i-1} | 0)}{\mathbf{W}_N^{(i)} = (y_1^N, \hat{u}_1^{i-1} | 1)} \quad (1)$$

根据式(1)得到的对数似然比进一步判决该比特的取值, 判决规则如式(2)

$$\hat{u}_i \triangleq \begin{cases} 0, & \mathbf{L}_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 0 \\ 1, & \text{其它} \end{cases} \quad (2)$$

在SCL译码算法中由于要保存 L 条备选路径, 因此根据文献^[15], 定义一个路径度量值(Path Metric, PM)

$$\text{PM}_i^l \triangleq \sum_{j=0}^i \ln \left(1 + e^{-(1-2\hat{u}_j^l) \cdot \mathbf{L}_N^{(j)l}} \right) \quad (3)$$

其中, l 表示路径索引值, \hat{u}_j^l 表示第 l 条路径中第

j 个比特的估计值。根据极化码特殊的构造, SC和SCL译码过程中的 $L_N^{(i)}$ 值可由文献[1]中式(75), 式(76)迭代计算, 本文中重写为式(4)和式(5)

$$L_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}) = \frac{L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}) + 1}{L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) + L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})} \quad (4)$$

$$L_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = \left[L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) \right]^{1-2\hat{u}_{2i-1}} \cdot L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}) \quad (5)$$

在实际应用中, 为了便于硬件实现, 式(3)的更新策略可替代为

$$PM_i^l = \begin{cases} PM_{i-1}^l, \hat{u}_i^l = \frac{1}{2} (1 - \text{sgn}(L_N^{(i)l})) \\ PM_{i-1}^l + |L_N^{(i)l}|, \text{其它} \end{cases} \quad (6)$$

在该更新策略中, 每一层路径扩展后都在不可靠路径的PM值上增加一个惩罚因子 $|L_N^{(i)l}|$, 因此在译码结束时, 选择PM值最小的 L 条路径输出。图1(a)给出当 $L=4$ 时SCL的部分译码过程, 每次扩展路径之后更新 PM_i^l 值, 每个父节点的左右分支分别表示该节点“0”和“1”的两种可能取值, 黑色实线表示经判决后保留的路径, 虚线为参与判决而被淘汰的路径, 点划线表示该阶段未参与任何计算的路径。图1(b)为SC算法译码树, 由于该算法仅保留1条路径, 因此不需要额外保存路径度量值, 只需计算判决, 根据 $L_N^{(i)}$ 判决得到 \hat{u}_i 值后直接进行下一阶段判决。

3 基于分段CRC的自适应SCL译码算法

3.1 AD-SCL算法与SCA-SCL算法

由极化码的SCL译码可知, 当设置的列表大小

L 越大时, 每一层更新后 PM_i^l 值的精度越高且该层保存的路径数越多, 在判决剪枝过程中使得正确路径存活概率更大。因此在低信噪比环境下采用较大 L 值, 可以提升译码性能, 而在高信噪比环境下采用较小的 L 值即可获得与采用较大 L 值相同的性能。AD-SCL算法将首次译码 L 值设为1, 即SC算法, 将接收码字全部译码后, 经过检验判断译码是否正确, 若不正确则扩大保存路径数 L , 再次进行译码, 直到译码正确或当前保存路径数等于预先设置的最大路径数 $L_{\max} = 2^m$ 。AD-SCL算法能够在高信噪比下采用较小的列表就能获得与采用较大列表相同的性能, 所以其能降低译码复杂度, 但是在信道环境较差的情况下, 需要遍历 $L=2, L=4, L=8, \dots, L=L_{\max}$ 每种情况下的整个译码过程, 这将导致较高的译码延迟。

SCA-SCL算法利用分段CRC, 判断部分译码序列是否正确。若此时译码树中有至少一条路径通过CRC校验, 则保留最可靠的一条路径继续译码, 否则反馈译码失败。当最后一段译码结束时, 若有至少一条路径通过CRC校验则只保留其中可靠度最高的一条路径作为译码输出, 否则反馈译码失败。该算法较CA-SCL算法, 可提前反馈译码是否正确, 在低信噪比下复杂度较CA-SCL更低, 高信噪比下增益不大。

因此, 本文提出一种基于分段CRC的自适应SCL算法(SCAD-SCL)。本算法结合了SCA-SCL算法中的分段译码和AD-SCL算法中保存径数动态变换的策略, 使得在整个信噪比区域上降低平均时延和复杂度, 并保证其性能达到 $L=L_{\max}$ 时的性能。

3.2 SCAD-SCL算法

SCAD-SCL算法基于SCA-SCL和AD-SCL算法, 在前一段译码结束时进行CRC校验, 如果校验错误则扩大列表大小 L 重新译码, 直到译码正确

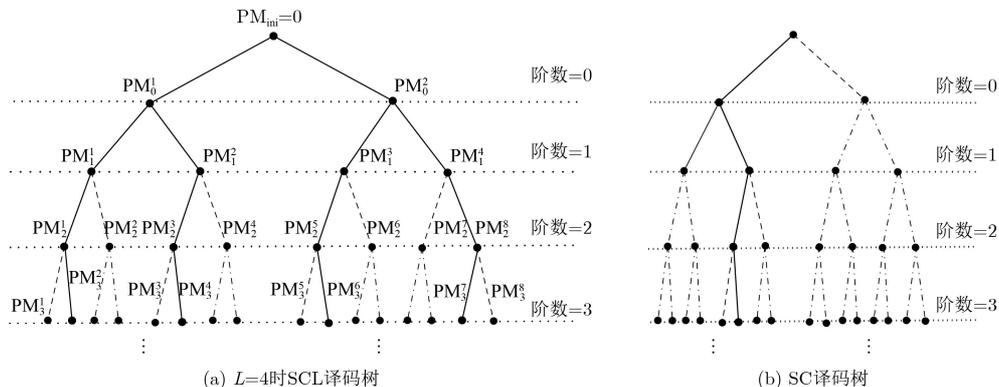


图1 SCL译码树与SC译码树

或者达到预设最大保留路径值，选择一条最可靠的路径保留输出，提高前半部分译码结果的可靠性的，再根据前一段的校验结果和列表大小 L 自适应选择后一段的列表大小 L ，该部分由 L 选择器完成如图2，图中“err”表示CRC校验结果，若通过校验则err = 0，否则err = 1。整个过程节省了错误路径扩展的无用计算，并在保留路径上进行扩展，完成第2部分自适应译码。具体算法流程图如图3。

根据算法流程图可知，该译码过程的顺利执行需要在极化码构造时对信息比特分段添加CRC。此外，由于极化码构造时，在比特之间引入的特殊结构，译码时第 i 个比特与前 $i - 1$ 个比特相关，因此在部分译码输出时，应当保留部分译码结果的参

数作为后续译码的输入参数，编译码具体实现步骤如下：

步骤 1 将长度为 K' 的信息比特分为两段，分别将前 $K'/2$ bit添加 X bit CRC，后 $K'/2$ bit添加 X bit CRC。将整段数据当做信息比特进行传输，此时信息比特长度为 $K = K' + X + X$ ；

步骤 2 生成置信序列，将冻结比特与信息比特混合后进行极化编码；

步骤 3 设置 $L = 2$ ，最大可扩展路径数为 L_{max} ，最小为 L_{min} ，初始化一条空路径；

步骤 4 利用SCL译码算法对接收比特进行译码。译至第 i bit时，若其中有 $K'/2 + X$ 位属于 A ，则提取 i 个译码比特中 $K'/2 + X$ 个信息位进行CRC校验。若至少有一条路径通过校验则保留其中 PM_i^l 值最小的路径作为部分译码输出，通过 L 选择器(如图3)设置 L 并执行步骤5。若校验失败则判断是否 $L < L_{max}$ ，是则通过 L 选择器并执行步骤4，否则选择当前译码路径中 PM_i^l 值最小的路径作为部分译码输出；

步骤 5 从 $(i + 1)$ bit继续译码，直到所有比特译码结束，进行第2次CRC校验，若至少有1条路径通过校验则保留其中 PM_i^l 值最小的路径作为部分译码输出，若校验失败且 $L < L_{max}$ ，则通过 L 选择器设置 L 并执行步骤5，如果 $L = L_{max}$ 则选择当前译码路径中 PM_i^l 值最小的路径作为部分译码输出；

步骤 6 将部分译码信息连接，并去除CRC，得到净信息比特，进行错误统计。

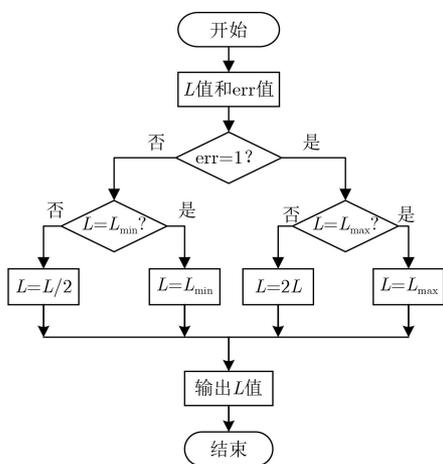


图 2 L选择器

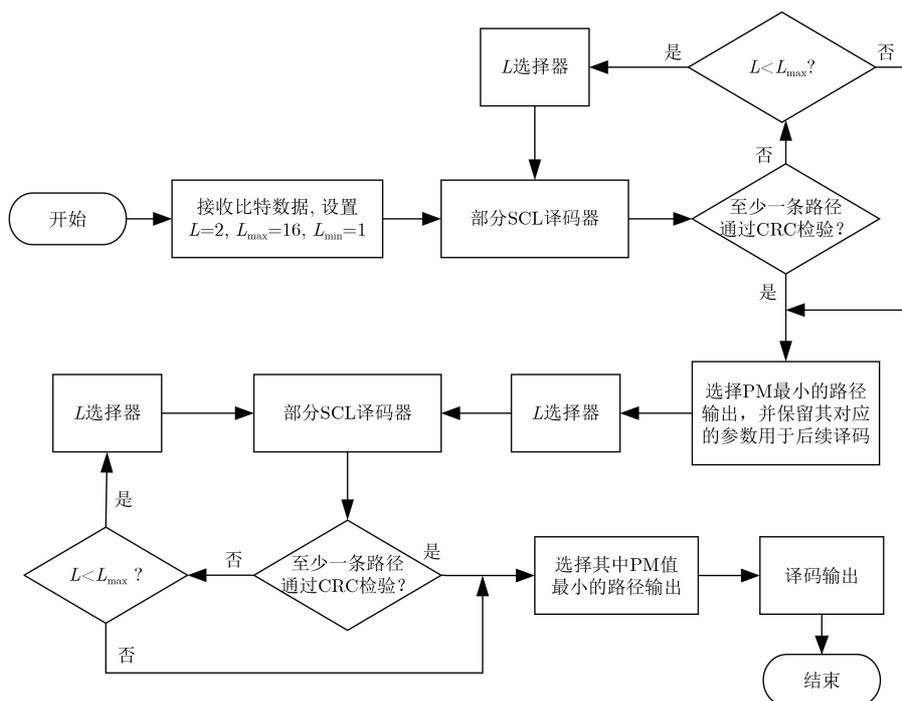


图 3 SCAD-SCL算法流程图

4 仿真分析

4.1 仿真参数设置

本文仿真是在高斯信道下进行, 码长 $N = 1024$, AD-SCL算法中采用32位的CRC, SCAD-SCL算法采用16位CRC, 由于其分段进行CRC添加, 总的冗余长度也为32 bit。假设两种算法的净信息比特数均为 K , 计算码率时, 将CRC比特也作为信息比特计算, 即 $R = (K+32)/N$ 。

在构造子信道置信序列时, 选择一个设计好的SNR值, 即文献[16]中提到的“designed SNR”, 该值通过蒙特卡洛仿真得到, 它能使构造出的置信序列在高斯环境下适用于一个范围内的SNR, 本文中选用designed SNR = -1.667进行仿真。

在设计 L 选择器时, 设置最大保留路径数 $L_{\max} = 16$, 最小保留路径数 $L_{\min} = 1$ 。当 $L = 1$ 时, 相当于使用SC算法, 译码复杂度低, 信道环境较好时, 能尽可能地降低译码复杂度。而选择16作为最大保留路径值是因为: 对比保留32条路径时的译码性能, 如图4, 可知 $L_{\max} = 16$ 与 $L_{\max} = 32$ 的译码性能相差不大, 但是复杂度 $L_{\max} = 32$ 是 $L_{\max} = 16$ 的2倍。综合考虑, 选择16作为仿真所用最大保留路径数。

本文中涉及的其他参数如表1所示。

4.2 结果分析

4.2.1 复杂度分析

表2所示当码率 $R = 0.5$ 时, 不同算法平均复杂度的比较。SCAD-SCL相比于CA-SCL算法, 当仿真环境设置码长为 N , CRC长度为 M 时, 每多一次校验, 多出的计算量为 $(N/2 - M) \times M$ 次加法。而与AD-SCL算法相比时, SCAD-SCL平均每次校验的计算量为 $(N/2 - M) \times M \times 2$ 次加法, AD-SCL每次校验的计算量为 $(N - 2M) \times 2M$ 次加法。当 $N > 32$ 时, AD-SCL的平均计算量均大于SCAD-SCL, 而论文仿真是在 $N = 1024$ 的情况下进行, 因此SCAD-SCL算法在CRC计算量这部分并不会高于AD-SCL算法。此外, CRC的计算量与译码器嵌套计算 $L_N^{(i)}$ 值相比, 可忽略不计。SCL类译码算法中计算复杂度最高的部分为迭代更新 PM_i^l 值, 因此本

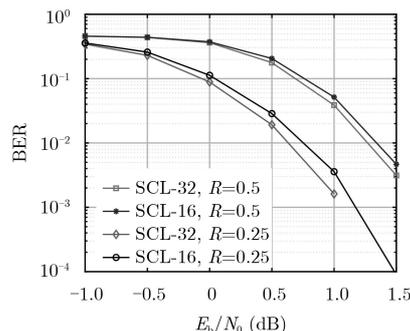


图4 SCL-16与SCL-32性能对比图

表1 仿真参数

仿真参数	具体内容
编码结构	$G_N = F^{\otimes n}$
信道环境	AWGN
调制方式	BPSK
子信道置信序列构造法	DE-GA
译码算法	CA-SCL, AD-SCL, SCAD-SCL

文中平均复杂度定义为: 更新 PM_i^l 值的次数。由表中数据可知, 在 $E_b/N_0 = -1$ dB时, SCAD-SCL较AD-SCL算法复杂度降低了约21.6%, 在高信噪比下($E_b/N_0 = 3$ dB)降低了约64%。

图5展示了在不同码率下SCAD-SCL译码算法复杂度与AD-SCL算法复杂度的对比。由图可知, 在不同码率下SCAD-SCL算法平均复杂度均低于AD-SCL算法的平均复杂度, 与始终保留16条路径译码的算法相比, 在高信噪比下优势则更大。

4.2.2 性能分析

图6、图7、图8分别给出在1/2, 1/4, 1/8码率下, CA-SCL算法(图例中用SCL-L表示, L 为列表大小)、AD-SCL算法与本文的SCAD-SCL算法的性能比较。由仿真结果可知, 在3种码率下, 本文的SCAD-SCL算法的译码性能与设置最大列表为16时的AD-SCL算法、SCL-16的性能几乎一致。

5 结论

针对AD-SCL算法每次完整译码后判决是否扩大保留路径重复译码导致的高复杂度和时延问题, 本文提出基于分段CRC的自适应SCL算法。通过部

表2 $R=0.5$ 时不同算法复杂度比较

	E_b/N_0 (dB)								
	-1.0	-0.5	0	0.5	1.0	1.5	2.0	2.5	3.0
AD-SCL	42304	42239	40850	32248	14153	4648	2034	1588	1536
SCAD-SCL	33177	33120	31918	24274	10590	2696	837	571	538

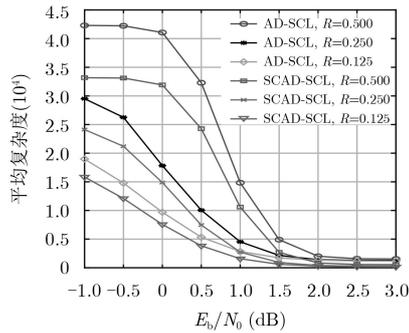


图 5 SCAD-SCL与AD-SCL复杂度对比

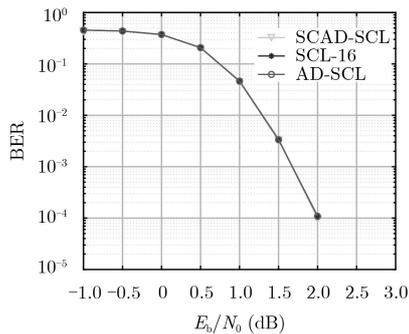


图 6 $R = 0.500$ 时, SCAD-SCL与AD-SCL误码性能比较

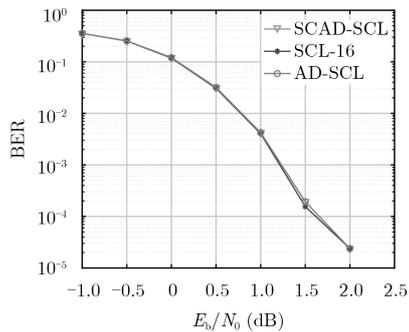


图 7 $R = 0.250$ 时, SCAD-SCL与AD-SCL误码性能比较

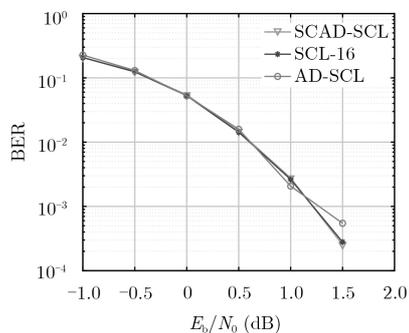


图 8 $R = 0.125$ 时, SCAD-SCL与AD-SCL误码性能比较

分译码后的校验值可实现提前终止错误的译码树,减少了无用的后续译码计算。通过仿真分析可知,在低信噪比情况下,SCAD-SCL算法能在降低平均计算复杂度和时延的前提下取得与AD-SCL相当的性能。

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. doi: 10.1109/TIT.2009.2021379.
- [2] ARIKAN E and TELATAR E. On the rate of channel polarization[C]. Proceedings of 2009 IEEE International Symposium on Information Theory, Seoul, South Korea, 2009: 1493–1495.
- [3] ZHANG Chuan and PARHI K K. Low-latency sequential and overlapped architectures for successive cancellation polar decoder[J]. *IEEE Transactions on Signal Processing*, 2013, 61(10): 2429–2441. doi: 10.1109/TSP.2013.2251339.
- [4] TAL I and VARDY A. List decoding of polar codes[C]. Proceedings of 2011 IEEE International Symposium on Information Theory Proceedings, St. Petersburg, Russia, 2011: 1–5.
- [5] ERCAN F, CONDO C, HASHEMI S A, *et al*. On error-correction performance and implementation of polar code list decoders for 5G[EB/OL]. <http://arxiv.org/abs/1708.04706>, 2017.
- [6] NIU Kai and CHEN Kai. CRC-aided decoding of polar codes[J]. *IEEE Communications Letters*, 2012, 16(10): 1668–1671. doi: 10.1109/LCOMM.2012.090312.121501.
- [7] ZHOU Huayi, ZHANG Chuan, SONG Wenqing, *et al*. Segmented CRC-aided SC list polar decoding[C]. Proceedings of the 2016 IEEE 83rd Vehicular Technology Conference, Nanjing, China, 2016: 1–5.
- [8] HASHEMI S A, CONDO C, and GROSS W J. Simplified successive-cancellation list decoding of polar codes[C]. Proceedings of 2016 IEEE International Symposium on Information Theory, Barcelona, Spain, 2016: 815–819.
- [9] HASHEMI S A, CONDO C, and GROSS W J. Fast simplified successive-cancellation list decoding of polar codes[C]. Proceedings of 2017 IEEE Wireless Communications and Networking Conference Workshops, San Francisco, USA, 2017: 1–6.
- [10] LI Bin, SHEN Hui, and TSE D. An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check[J]. *IEEE Communications Letters*, 2012, 16(12): 2044–2047. doi: 10.1109/LCOMM.2012.111612.121898.
- [11] MORI R and TANAKA T. Performance of polar codes with the construction using density evolution[J]. *IEEE*

- Communications Letters*, 2009, 13(7): 519–521. doi: [10.1109/LCOMM.2009.090428](https://doi.org/10.1109/LCOMM.2009.090428).
- [12] WU Daolong, LI Ying, and SUN Yue. Construction and block error rate analysis of polar codes over AWGN channel based on Gaussian approximation[J]. *IEEE Communications Letters*, 2014, 18(7): 1099–1102. doi: [10.1109/LCOMM.2014.2325811](https://doi.org/10.1109/LCOMM.2014.2325811).
- [13] SCHURCH C. A partial order for the synthesized channels of a polar code[C]. Proceedings of 2016 IEEE International Symposium on Information Theory, Barcelona, Spain, 2016: 220–224.
- [14] HE Gaoning, BELFIORE J C, LAND I, *et al.* Beta-expansion: a theoretical framework for fast and recursive construction of polar codes[C]. Proceedings of 2017 IEEE Global Communications Conference, Singapore, 2017: 1–6.
- [15] BALATSOUKAS-STIMMING A, PARIZI M B, and BURG A. LLR-based successive cancellation list decoding of polar codes[C]. Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 2014: 3903–3907.
- [16] VANGALA H, VITERBO E, and HONG Yi. A comparative study of polar code constructions for the AWGN channel[EB/OL]. <http://arxiv.org/abs/1501.02473>, 2015.
- 王 琼: 女, 1971年生, 高级工程师, 研究方向为移动通信.
- 罗亚洁: 女, 1994年生, 硕士生, 研究方向为移动通信物理层算法、信道编码.
- 李思航: 男, 1992年生, 硕士生, 研究方向为移动通信物理层算法、信道编码.