

DOI:10.11918/j.issn.0367-6234.201706116

# 低复杂度极化码 SCL 译码算法

刘士平, 马林华, 胡 星, 黄天宇

(空军工程大学 航空航天工程学院, 西安 710038)

**摘要:** 极化码的串行抵消列表(SCL)译码的逐次逐比特进行判决过程与路径度量值的计算筛选过程是整个译码系统复杂度与延迟的主要来源. 在分析现有 SCL 及多比特判决译码的优缺点基础上, 针对 SCL 译码造成的译码系统复杂度高和延时大的问题, 将每组多比特码字(多个独立信道)视作一个整体, 并在译码时通过信道合成构造为一个虚拟多比特信道, 从而可以对多比特码字进行同步传输并根据信道递归公式进行同步判决译码. 由此基于 SCL 译码的码树构造提出一种构造多比特虚拟信道的 SCL 译码算法, 并结合设置译码码树节点阈值减少码树节点分裂次数的方法进一步增强了上述算法. 在 AWGN 信道下的分别对虚拟 2、4 和 8 比特信道 SCL 译码的误码率及误帧率性能进行仿真. 仿真结果表明在虚拟 8 比特信道情况下, 预设阈值  $S = 30$  时的译码性能接近传统 SCL 算法, 且总历经节点数降低了 63.7%, 总加法次数是 8 比特同时判决译码算法的 17%. 此算法降低了译码算法的计算复杂度及硬件存储复杂度, 更适合于硬件实现, 具有一定的实用价值.

**关键词:** 极化码; 串行抵消列表译码; 译码复杂度; 阈值

中图分类号: TN911.22

文献标志码: A

文章编号: 0367-6234(2018)05-0160-06

## Alow complexity SCL decoding algorithm for polar codes

LIU Shiping, MA Linhua, HU Xing, HUANG Tianyu

(Institute of Aeronautics and Astronautics Engineering, Air Force Engineering University, Xi'an 710038, China)

**Abstract:** The successive cancellation list (SCL) decoding algorithm of polar code is of bitwise, which is the main source of the decoding complexity and latency. In view of the analysis of the advantages and disadvantages of the existing SCL and multi-bit decision decoding, each set of multi-bit code words (multiple independent channels) is treated as a whole to construct a multi-bit virtual channel, which can transmit multi-bit code words and decode synchronously. Therefore a construction of multi-bit virtual channel based on the SCL algorithm under the AWGN channel is proposed to against the problem of high complexity and large delay for decoding system caused by SCL decoding. It is also enhanced by combining with the establishment threshold of the decoder tree nodes to reduce the number of tree nodes' split. Under the AWGN channel, the bit error rate and the frame error rate performance of the virtual 2, 4 and 8 bit channels are simulated. The simulation results show that the decoding performance is close to that of the traditional SCL algorithm when the default threshold  $S = 30$  in the case of a virtual 8-bit channel. The total number of nodes is reduced by 63.7% and the total number of additions is 17% of the 8-bit decision decoding algorithm. The proposed algorithm can effectively reduce the calculation complexity of decoding and the hardware storage. It is more suitable for hardware implementation and has a certain practical value.

**Keywords:** polar code; successive cancellation list algorithm; decoding complexity; threshold

目前, 香农在有噪信道编码理论中指出, 存在可以达到香农限的码字<sup>[1]</sup>. 极化码(Polar Code)是 2008 年由 E. Arıkan 提出的一种基于信道极化定理进行设计的新型信道编码, 是第一种能够通过严格的数学方法证明达到信道容量的构造性编码方案<sup>[2-5]</sup>. 因此极化编码激起了各国学者的广泛研究

并开始应用于各个领域, 拥有良好的发展前景.

串行抵消(SC)译码算法是极化码的基本译码算法, 当码长趋近于无穷时, 可获得很好的渐进性能<sup>[2]</sup>. 但是在中短码长时, SC 译码的性能并不理想, 相同条件下的编码增益不如 Turbo 码及 LDPC 码<sup>[6]</sup>. 为提高 SC 译码性能, 文献提出了串行抵消列表(SCL)算法, 增加了每层译码搜索宽度, 降低了误码率. 在此基础上, 文献[9-10]提出基于对数似然比(LLR)的 SCL 译码算法, 对译码码树中每一个节点的判决计算次数缩减了一半, 降低了计算及硬件存储复杂度. 但上述两种方法都是逐比特串行译码, 运算量及运算复杂度随着码长的增加呈指数增长. 针对此类问题, 文献[11-13]中多比特译码算

收稿日期: 2017-06-19

基金项目: 国家自然科学基金资助项目(61472442); 综合业务网国家重点实验室(西安电子科技大学)开放研究课题资助项目(INS-15-13); 航空科学基金资助项目(20155896025)

作者简介: 刘士平(1994—), 男, 博士研究生;  
马林华(1965—), 男, 教授, 博士生导师.

通信作者: 马林华, Land\_max@126.com

法,将传统 SCL 的单比特译码增强为多比特同时译码. 此算法尽管减少了译码器判决次数降低了译码延时,但由于其译码算法结构,随着每次译码的比特数增加,运算次数及硬件存储复杂度都会随之增加. 基于上述问题,本文提出了基于构造虚拟多比特信道的 SCL 译码算法. 将传统的多个单比特信道构造成多比特虚拟信道,降低了译码运算及存储复杂度,并结合文献[14]中提出的基于信道可靠度的阈值判决的译码方案,进一步增强了此算法,提高译码效率.

## 1 背景算法描述

### 1.1 极化码 SCL 译码算法

$(N, K)$  极化编码输入的  $N$  比特信息  $u_1^N = (u_1, u_2, \dots, u_n)$  由  $K$  比特的信息比特和  $(N - K)$  位冻结比特(一般为全 0)两部分组成. 根据信道极化理论,分别对  $N = 2^n$  个独立的二进制离散无记忆信道(B-DMC)  $W$  进行信道合成及信道分解的操作后,可得到  $N$  个独立等价子信道  $W_N^{(i)}, i = 1, 2, \dots, N$ . 其中一部分子信道的对称容量  $I(W_N^{(i)})$  接近于 1, 视作无噪信道,用于传输信息比特;而另一部分子信道的对称容量  $I(W_N^{(i)})$  接近于 0, 视作全噪信道,用于传输冻结比特.

根据文献[2]中提到的有关信道可靠性的计算方法,  $u_1^N$  中的  $K$  比特消息通过可靠性高的子信道进行传输,而  $(N - K)$  位冻结比特通过可靠性低的子信道进行传输. 其次,依据编码式(1),完成极化编码,其中  $G_N$  为生成矩阵.

$$x_1^N = u_1^N G_N = u_A G_N(A) \oplus u_{A^c} G_N(A^c). \quad (1)$$

式中:  $A$  为信息信道下标的集合;  $A^c$  为其补集即冻结信道的下标集合;符号  $\oplus$  为模二加运算. 生成矩阵  $G_N$  为

$$G_N = B_N F^{\otimes n}. \quad (2)$$

式中:符号  $\otimes$  为张量积运算;  $B_N = R_N(I_2 \otimes B_{N/2})$ , 实现索引位逆转操作;  $R_N$  是比特翻转矩阵即实现元素的奇偶分离重排.

由此可见,极化码也是线性分组码的一种,  $(N, K)$  极化码的码率  $R = N/K$ . 极化编码可视作根据信道可靠度的计算选取合适的信道,确定编码参数  $(N, K, A, u_{A^c})$  的  $G_N$  陪集编码过程.

在仿真实验中,极化码的码字序列经二进制相移键控(BPSK)调制为符号  $x_1^N \in \{-1, 1\}^N$  的序列,经过 AWGN 信道后得到的接收序列  $y_1^N = (y_1, y_2, \dots, y_N)$ . 采用串行抵消(SC)译码可看作逐个依次地根据接收端计算出的对数似然比(LLR)进行硬判

决.  $u_i$  的似然函数为  $W_n^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i)$ , LLR 为

$$L(u_i) = \ln \left( \frac{W_n^{(i)}(y_1^N, \hat{u}_1^{i-1} | 0)}{W_n^{(i)}(y_1^N, \hat{u}_1^{i-1} | 1)} \right). \quad (3)$$

式中  $\hat{u}_1^{i-1}$  是接收端对前  $i - 1$  个比特的估计序列. 根据上述得到的 LLR 值结合硬判决即可完成对  $u_i$  的估计,具体判决依据如下:

$$\hat{u}_i = \begin{cases} 0, & i \in A, L(u_i) > 0; \\ 1, & i \in A, L(u_i) < 0; \\ u_i, & i \in A^c. \end{cases} \quad (4)$$

在上述基于 LLR 的 SC 译码算法的基础上,根据译码码树,若在每次按位搜索时将搜索宽度扩大为  $L$ ,即保留  $L$  条候选路径. 然后基于  $L$  条路径依次扩展,直到叶子节点即最后一层. 从最后一层保留的  $L$  条路径中选择一个路径度量值最大的路径作为译码输出,此种方法即为基于 LLR 的 SCL 译码算法. 因此, SCL 算法可看作是多路径同时进行的 SC 译码算法.

### 1.2 多比特判决译码算法

在极化码的译码算法中,路径度量的判决筛选过程是整个系统延迟的主要来源. 无论是 SC 译码还是 SCL 译码算法都依据硬判决式(4)逐次逐位进行判决. 当码长增加时,该算法的系统延迟增加,降低了译码效率. 针对此问题,文献[13]提出了多比特判决的译码算法( $2^k b - r$ SCL 算法),译码时可每次同时计算出  $M$  比特消息的路径度量值进行判决筛选. 如图 1 所示,当  $M = 1$  时等价于传统 SCL 译码,当  $M$  时,该算法即为最大似然(ML)译码.

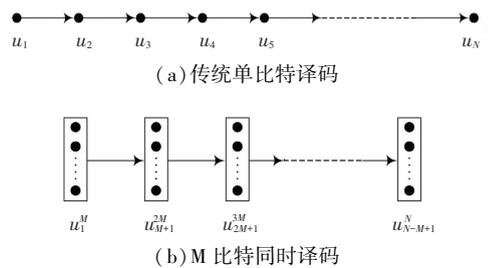


图 1 传统单比特及多比特译码示意

Fig.1 Traditional singal-bit and multi-bit decoding schematic

假设  $W: X \rightarrow Y$  为一个 B-DMC 信道,则  $N$  个并列单比特信道为  $W_N^{(i)}: X \rightarrow Y^N \times X^{i-1}, 1 \leq i \leq N$ . 在多比特同步译码算法中,  $M(M = 2^k)$  比特码字的路径可靠性度量过程视作由  $M$  个独立的单比特信道进行信道合成得到,即  $W_{N,M}^{(j)}: X^M \rightarrow Y^N \times X^{jM-M}, 1 \leq j \leq \frac{N}{M}$ , 其转移概率为  $W_{N,M}^{(j)}(y_1^N, x_1^{jM-M} | x_{jM-M+1}^{jM})$ . 基于信道与信道的输出码字  $u_1^N$  的独立性,上述算法可

描述为:当  $M$  比特码字同时译码判决时,信道转移概率  $W_{N,M}^{(i)}(y_1^N, \hat{u}_1^{iM-M} | u_{iM-M+1}^{iM})$  转化成  $M$  个独立的单比特信道之积,即

$$W_{N,M}^{(i)}(y_1^N, \hat{u}_1^{iM-M} | u_{iM-M+1}^{iM}) = \prod_{j=0}^{M-1} W_{\frac{N}{M}}^{(i)}(y_{\frac{N}{M}+1}^{(j+1)\frac{N}{M}}, \hat{\alpha}_{1+\frac{N}{M}}^{(i-1)+\frac{jN}{M}} | \alpha_{i+\frac{jN}{M}}). \quad (5)$$

式中  $u_{iM-M+1}^{iM} = (\alpha_i, \alpha_{i+\frac{N}{M}}, \dots, \alpha_{i+N-\frac{N}{M}}) \mathbf{G}_M$ ; 生成矩阵  $\mathbf{G}_M = \mathbf{B}_M \mathbf{F}^{\otimes K}$ . 根据得到的似然值  $W_{N,M}^{(j)}(y_1^N, \hat{u}_1^{iM-M} | u_{iM-M+1}^{iM})$ , 结合 SCL 译码算法,通过路径度量模块 (MCU) 进行筛选保留  $L$  条候选路径以及通迫零算法模块 (ZFU) 将冻结位至“0”. 通过增加了每次译码比特数从而提高译码效率.

## 2 基于虚拟多比特信道的 SCL 译码

多比特译码算法的核心是对似然值  $W_{N,M}^{(i)}(y_1^N, \hat{u}_1^{iM-M} | u_{iM-M+1}^{iM})$  的计算. 在对图 1(b) 中每组  $M$  比特码字进行译码时,上述算法将其转化为传统单比特信道似然值之积. 然而该算法只是通过路径度量上的多位判决降低了译码延迟,随着码长以及  $M$  的增加,计算复杂度以及进行路径度量判决 (MCU) 的查表复杂度也随之增加. 并没有从信道转移概率这个根本问题上解决多比特运算的运算及存储复杂度.

将每组  $M$  比特码字即  $M$  个独立信道视作一个整体. 在进行译码判决时,可将  $M$  个独立单比特信道通过信道合成构造为一个独立的  $M$  比特虚拟信道. 此虚拟信道可以同时为  $M$  比特码字进行传输并根据信道递归公式进行同步判决,而不是  $M$  个单比特信道之积. 这个虚拟信道为  $W_N^{(i)}: \mathbf{A} \rightarrow \mathbf{Y}^N \times \mathbf{A}^{i-1}$ ,  $1 \leq i \leq \frac{N}{M}$ , 信道转移概率为  $W_n^{(i)}(y_1^N, \lambda_1^{i-1} | \lambda_i)$ ,  $\lambda_i = (u_{iM-M+1}, \dots, u_{iM})$ . 根据虚拟信道的性质,对上述转移概率  $W_{N,M}^{(i)}(y_1^N, \hat{u}_1^{iM-M} | u_{iM-M+1}^{iM})$  的计算可通过信道递归计算有进一步地简化,并有下述定理 1.

**定理 1** 假设  $u_1^N$  中每一个比特  $u_i$  分别独立的等概率的判决为 0 或 1,那么给定  $0 < m \leq n, N = 2^n, M = 2^m$ , 对任意的  $1 < k \leq m, 0 \leq \omega < n$ , 令  $K = 2^k, \Omega = 2^\omega$ , 并且  $0 \leq i < \frac{2\Omega}{K}$ , 则  $K$  比特虚拟信道

$W_{2\Omega,K}^{(i+1)}$  可以由两个独立的虚拟  $\frac{K}{2}$  比特信道  $W_{\Omega,K/2}^{(i+1)}$  通过单步信道变换得到,即

$$(W_{\Omega,K/2}^{(i+1)}, W_{\Omega,K/2}^{(i+1)}) \mapsto W_{2\Omega,K}^{(i+1)}$$

且信道转移概率满足

$$W_{2\Omega,K}^{(i+1)}(y_1^{2\Omega}, u_1^{iK} | u_{iK+1}^{iK+K}) =$$

$$W_{\Omega,K/2}^{(i+1)}(y_1^\Omega, u_{1,o}^{iK} \oplus u_{1,e}^{iK} | u_{iK+1,o}^{iK+K} \oplus u_{iK+1,e}^{iK+K}) \cdot W_{\Omega,K/2}^{(i+1)}(y_{\Omega+1}^{2\Omega}, u_{1,e}^{iK} | u_{iK+1,e}^{iK+K}). \quad (6)$$

**证明** 由条件概率公式  $\Pr(B|A) = \Pr(A, B)/\Pr(A)$  可得

$$W_{2\Omega,K}^{(i+1)}(y_1^{2\Omega}, u_1^{iK} | u_{iK+1}^{iK+K}) = \frac{W_{2\Omega,1}^{(iK+K)}(y_1^{2\Omega}, u_1^{iK+K-1} | u_{iK+K})}{\Pr(u_{iK+1}^{iK+K-1} | u_{iK+K})}$$

由于每一个比特  $u_i$  分别独立的等概率的判决为 0 或 1,则

$$\Pr(u_1^{iK+K-1} | u_{iK+K}) = \Pr(u_1^{iK+K-1}) = 2^{-(K-1)}$$

因此

$$W_{2\Omega,K}^{(i+1)}(y_1^{2\Omega}, u_1^{iK} | u_{iK+1}^{iK+K}) = 2^{(K-1)} W_{2\Omega,K}^{(i+1)}(y_1^{2\Omega}, u_1^{iK+K-1} | u_{iK+K}). \quad (7)$$

根据文献 [2] 中的式 (23) 可得

$$W_{2\Omega,1}^{(iK+K)}(y_1^{2\Omega}, u_1^{iK+K-1} | u_{iK+K}) = \frac{1}{2} W_{\Omega,1}^{(iK+K)}(y_1^\Omega, u_{1,o}^{iK+K-2} \oplus u_{1,e}^{iK+K-2} | u_{iK+K-1} \oplus u_{iK+K}) \cdot W_{\Omega,1}^{(iK+K)}(y_{\Omega+1}^{2\Omega}, u_{1,e}^{iK+K-2} | u_{iK+K}) \quad (8)$$

可以推出

$$W_{\Omega,1}^{(iK+K)}(y_1^\Omega, u_{1,o}^{iK+K-2} \oplus u_{1,e}^{iK+K-2} | u_{iK+K-1} \oplus u_{iK+K}) = 2^{-(\frac{K}{2}-1)} W_{\Omega,K/2}^{(i+1)}(y_1^\Omega, u_{1,o}^{iK} \oplus u_{1,e}^{iK} | u_{iK+1,o}^{iK+K} \oplus u_{iK+1,e}^{iK+K}) \quad (9)$$

同理

$$W_{\Omega,1}^{(iK+K)}(y_{\Omega+1}^{2\Omega}, u_{1,e}^{iK+K-2} | u_{iK+K}) = 2^{-(\frac{K}{2}-1)} W_{\Omega,K/2}^{(i+1)}(y_{\Omega+1}^{2\Omega}, u_{1,e}^{iK} | u_{iK+1,e}^{iK+K}) \quad (10)$$

综上,由式 (7) ~ (10), 即可推出 (6), 证毕.

当利用上述多比特虚拟信道算法进行译码时,译码树每层对  $K$  位信息  $u_{iK+1}^{iK+K}$  的判决有  $2^K$  条候选路径,因此基于 LLR 的二元判决模式不再实用. 针对此问题,结合上述基于阈值度量的算法采取了对数似然 (LL) 结构,由上述定理 1 可得到每条候选路径的 LL 值即  $\log W_{2\Omega,K}^{(i+1)}(y_1^{2\Omega}, u_1^{iK} | u_{iK+1}^{iK+K})$ .

随着虚拟信道数  $K$  的增加,译码判决候选路径数  $2^K$  也呈指数增长. 为解决此问题,引入 AWGN 信道下基于高斯近似的低分裂次数优化算法,进一步增强上述译码算法,降低路径度量运算以及存储复杂度. 文献 [14] 提出的 ESR-SCL 译码算法中,引入  $P_e(u_i)$  为在译码过程中节点  $u_i$  的平均错误概率,即在子信道  $W_n^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i)$  中,当  $u_1^{i-1}$  判决全部正确时  $u_i$  被错误译码判决的概率. 则  $1 - P_e(u_i)$  为  $u_i$  节点的译码可靠性度量值即置信度,并作为译码节点的阈值.  $P_e(u_i) = Q(\sqrt{E[L(u_i)]/2})$ , 其中  $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-\frac{t^2}{2}} dt$ . 根据阈值的预先计算提出了

上述算法的核心,即只有当  $W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i = 0) > 1 - P_e(u_i)$  或  $W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i = 1) > 1 - P_e(u_i)$  时,译码码树在此层不分裂直接判为 0 或 1,从而减少了历经节点数,降低了复杂度。

由高斯近似可知,当任意  $i \in [1, N]$  时,  $u_i$  取 1 或 0 的  $L(u_i)$  值具有相同的概率密度函数<sup>[15]</sup>。因此  $K$  比特信息  $u_{iK+1}^{iK+K}$  的置信度为  $(1 - P_e(u_i))^K$ , 其中  $iK + 1 \leq j \leq iK + K$ 。然而由上述公式,  $P_e(u_i)$  中变量  $L(u_i)$  的计算需要得到已经译码判决出的  $u_1^{i-1}$ 。根据多比特虚拟信道的译码码树构造可知,在对  $u_{iK+1}^{iK+K}$  进行判决时,  $\hat{u}_1^{iK}$  的判决结果是已知的,故只有  $u_{iK+1}$  的 LLR 值  $L(u_{iK+1})$  可以通过计算得到。因此上述计算中取  $j = iK + 1$ , 则  $u_{iK+1}^{iK+K}$  置信度可定义为  $(1 - P_e(u_{iK+1}))^K$ 。其中:

$$P_e(u_{iK+1}) = Q(\sqrt{E[L(u_{iK+1})]}/2),$$

$$L(u_{iK+1}) = \log \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{iK} | 0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{iK} | 1)} \quad (11)$$

结合虚拟信道的信道转移概率 LL 值以及上述给出的可靠性度量的阈值算法,定义通过上述虚拟信道传输算法译码时只有当  $\log(W_{2\Omega, K}^{(i+1)}(y_1^{2\Omega}, u_1^{iK} | u_{iK+1}^{iK+K} = \psi_{iK+1}^{iK+K})) > K \log(1 - P_e(u_{iK+1}))$  时,译码码树在此层不分裂直接判为  $\psi_{iK+1}^{iK+K}$ , 其中  $\psi_{iK+1}^{iK+K}$  为  $u_{iK+1}^{iK+K}$  的  $2^K$  种可能码字中任意一个码字。其余情况下结合多比特虚拟信道 SCL 译码算法正常分裂。即

$$\hat{u}_{iK+1}^{iK+K} = \begin{cases} \psi_{iK+1}^{iK+K}, \log(W_{2\Omega, K}^{(i+1)}(y_1^{2\Omega}, u_1^{iK} | u_{iK+1}^{iK+K} = \psi_{iK+1}^{iK+K})) \geq K \log(1 - P_e(u_{iK+1})) \\ \text{split, else.} \end{cases} \quad (12)$$

基于可靠性度量的计算可以推断,在路径正常分裂后,若出现  $\log(W_{2\Omega, K}^{(i+1)}(y_1^{2\Omega}, u_1^{iK} | u_{iK+1}^{iK+K} = \psi_{iK+1}^{iK+K})) \leq K \log(P_e(u_{iK+1}))$  则该路径默认为错误路径不再保留或扩展,直接删除。

基于上述算法的译码路径可分为两类,第一类是基本上不分裂,此类路径以很高的概率包含正确的码字;第二类是进行频繁地分裂,此类路径译码正确的概率相对较低。因此,定义变量  $S_l(i)$  为第  $l$  条路径在到达译码树第  $i$  层时中间经历的分裂次数。即假如第  $l$  条路径在第  $i$  层扩展到第  $i + 1$  层时没有分裂,那么  $S_l(i + 1) = S_l(i) + 1$ 。正确的译码路径很少分裂,而错误的译码路径会频繁地分裂,定义阈值  $S$ 。当  $S_l(i) \geq S$ , 则此路径很大可能为正确译码,予以保留,其余  $S_l(i) < S$  的路径予以抛弃,从而减少保留的路径数量,以较慢的速度达到 SCL 译码中定义的  $L$  值。

综上所述,提出基于阈值判决的多比特虚拟信道 SCL 译码算法,描述步骤如下:

第 1 步:根据式(6)的计算,得到译码输出每组  $K$  比特信息的 LL 值,共计  $2^k$  个值。

第 2 步:将上一步得到的  $2^k$  个 LL 值依据式(12)和(13)对其进行判决,筛选出不分裂的、被删除的以及正常分裂的子节点,  $S_l(i)$  值也随时保持更新。

第 3 步:在第 2 步筛选基础上,若此时保留路径数量  $P > L$ , 则删除  $S_l(i) < S$  的路径;若所有路径  $S_l(i) < S$ , 或删除  $S_l(i) < S$  的路径后  $P$  仍大于  $L$ , 则保留剩余路径中路径度量值最大的  $L$  条。

第 4 步:若  $i < \frac{N}{K}$ , 则令  $i = i + 1$  并重复第一步至第 3 步;若  $i = \frac{N}{K}$ , 则在最后一层保留路径中取出路径度量值最大的一条路径作为译码输出。

### 3 仿真结果及性能分析

在 AWGN 信道下选取码长为 1024, 码率为 1/2 的 (1024, 512) 极化码, 经过二进制频移键控 (BPSK) 调制, 分别对上述多比特虚拟信道的 SCL 译码及基于不同阈值判决的虚拟信道 SCL 译码算法进行 1 000 次蒙特卡洛仿真。

图 2 为 (1024, 512) 极化码的传统的 SCL 译码算法与  $K$  分别取 2、4、8 的多比特虚拟信道 SCL 译码算法的误码率及误帧率的对比。从 FER/BER 曲线可以看出, 当虚拟信道的比特数增大时, 该算法的译码性能由略微的增加但是并不明显, 性能与传统的 SCL 译码算法基本保持一致。

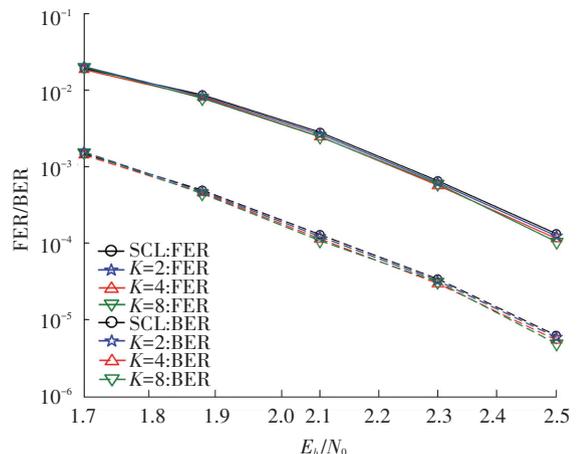


图 2 (1 024, 512) 极化码不同参数下 SCL 译码算法译码性能的对比如

Fig.2 The comparison of SCL decoding performance of (1024, 512) polar code under different parameters

以 4 比特同时译码为例,由式(6)对  $\log(W_{N,4}^{(j)}(y_1^N,$

$\hat{u}_{4i-4}^{4i-4} | u_{4i-4+1}^{4i}$ ) 的计算共需要经历  $(2^4 + 2 \times 2^2)$  共计 24 次加法运算, 而由式(5)对  $\log(W_{N,4}^{(j)}(y_1^N, \hat{u}_{4i-4}^{4i-4} | u_{4i-4+1}^{4i}))$  的计算需进行  $(2^4 \times (4 - 1))$  共计 48 次加法运算, 以此类推, 得到如下复杂度对比表.

表 1 计算  $\log(W_{2L,K}^{(i+1)}(y_1^{2L}, u_1^{iK} | u_{iK+1}^{iK+K}))$  所需加法次数

Tab.1 Numbers of additions to calculate  $\log(W_{2L,K}^{(i+1)}(y_1^{2L}, u_1^{iK} | u_{iK+1}^{iK+K}))$

K 值	虚拟信道		本文算法 优化率%
	算法	算法	
K = 2	4	4	0
K = 4	24	48	50
K = 8	304	1 792	83
K = 16	66 144	1 179 648	94.4

由图 2 和表 1 的分析可得出, 上述多比特虚拟信道的译码算法与文献[13]中多比特判决译码相比, 在  $K = 4$  与  $K = 8$  时可分别减少 50% 及 83% 的加法运算次数. 在降低了译码计算复杂度的同时, 译码性能与传统 SCL 算法基本保持一致.

图 3、4 为当  $(1\ 024, 512)$  极化码多比特虚拟信道译码时的  $K$  值分别取 4、8 时, 在不同译码码树节点阈值  $S$  下的译码性能曲线. 由不同阈值下的对比可知, 根据阈值判决的虚拟比特信道译码由于减少了码树分裂次数即历经节点数导致译码性能略差. 当所选取的阈值越大越接近传统 SCL 译码或虚拟比特信道 SCL 译码, 随着阈值  $S$  的减小, 译码性能也随之下降. 在阈值  $S$  达到 30 时, 性能略优于文献[14]中的 ESR-SCL 译码算法,  $S = 20$  和 25 时译码性能较 ESR-SCL 算法略差.

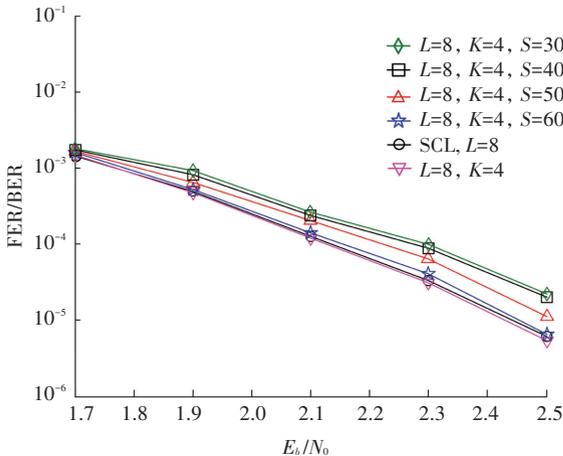


图 3  $(1\ 024, 512)$  极化码虚拟 4 比特信道译码时的不同阈值  $S$  下的译码性能

Fig.3 The decoding performance of  $(1\ 024, 512)$  polar code at different thresholds  $S$  under 4-bit virtual channel

由上述实验得到, 在  $K = 8$ 、信噪比为 2.5 dB 时误码率可达到  $10^{-5}$ . 图 5 为当信噪比  $E_b/N_0 = 2.5$  dB

时, 无预设阈值与不同阈值下虚拟 8 比特信道 SCL 译码码树每层历经的节点总数. 从图 3、4 观察对比可以发现, 设定阈值后, 虽然译码性能略有下降, 但每层历经的节点数量明显减少, 在图 5 中可观察不同阈值下历经节点数减少的具体情况. 表 3 记录了信噪比  $E_b/N_0 = 2.5$  dB 时不同参数下 1 000 次实验译码树平均历经的节点总数.

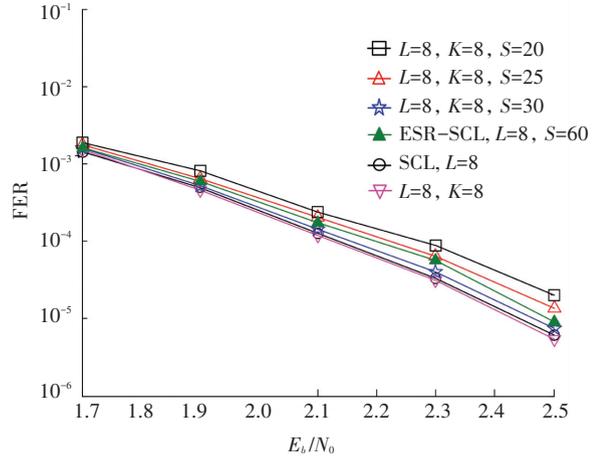


图 4  $(1\ 024, 512)$  极化码虚拟 8 比特信道译码时的不同阈值  $S$  下的译码性能

Fig.4 The decoding performance of  $(1\ 024, 512)$  polar code at different thresholds  $S$  under 8-bit virtual channel

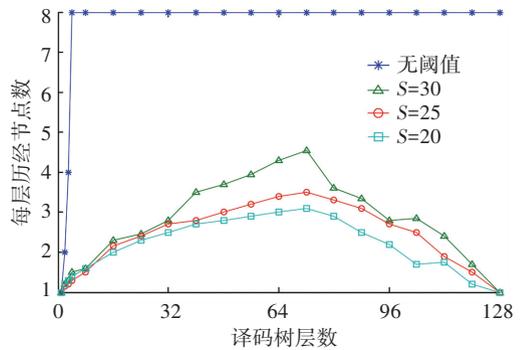


图 5  $(1\ 024, 512)$  极化码虚拟 8 比特信道译码时的不同阈值  $S$  下的平均历经节点总数

Fig.5 The decoding performance of  $(1\ 024, 512)$  polar code at different thresholds  $S$  under 8-bit virtual channel

表 2 不同阈值参数平均历经节点总数

Tab.2 The average experience nodes under different thresholds

S 值	无	30	25	20
历经节点数	1007	365.13	319.00	289.07

仿真实验表明, 当阈值  $S = 30$  时, 虚拟 8 比特信道 SCL 译码性能接近于传统 SCL 译码, 但是总历经节点数降低了 63.7%, 所需加法计算次数与 8 比特同时判决译码算法相比减少了 83%. 降低了译码算法的复杂度及历经节点次数, 降低了硬件存储复

复杂度.

## 4 结 论

本文回顾了极化码编码及 SCL 译码的原理,分析论证了现有多比特判决译码的优缺点.在此基础上提出了基于构造多比特虚拟信道的 SCL 译码算法并给出了多比特虚拟信道的递归公式,并结合上述算法的译码码树的构造结合了基于节点阈值判决的减少译码树分裂次数的算法,最后对上述算法进行了仿真实验与复杂度分析.结果表明,采用虚拟 8 比特信道 SCL 译码,取预设阈值  $S = 30$  时译码性能接近传统 SCL 算法,在信噪比  $E_b/N_0 = 2.5$  dB 时误码率可达到  $10^{-5}$ . 总历经节点数降低了 63.7%,总加法计算次数与 8 比特同时判决译码算法相比减少了 83%,降低了译码算法的计算复杂度及硬件存储复杂度,因此,该算法更适合于硬件实现,更有应用价值.

## 参考文献

- [1] SHANNON C E. A mathematical theory of communication [J]. Bell System Technical Journal, 1948, 19(4): 271–285. DOI: 10.1002/j.1538-7305.1948.tb01338.x.
- [2] ARICAN E. Channel polarization: a method for constructing capacity achieving codes for symmetric binary-input memoryless channels [J]. IEEE Trans. Inform Theory, 2009, 55(7): 3051–3073. DOI: 10.1109/TIT.2009.2021379.
- [3] ARICAN E. A performance comparison of polar codes and Reed-Muller codes [J]. IEEE Communications Letters, 2008, 12(6): 447–449. DOI: 10.1109/LCOMM.2008.080017.
- [4] ARICAN E. Channel combining and splitting for cutoff rate improvement [J]. IEEE Trans. Inf. Theory, 2006, 52(2): 628–639. DOI: 10.1109/ISIT.2005.1523420.
- [5] ARICAN E, KIM H, MARKARIAN G. Performance of short polar codes under ML decoding [C] // Paul Cunningham and Miriam Cunningham (Eds), IIMC International Information Management

Corporation, 2009: 1–6.

- [6] CHEN Kai, NIU Kai and LIN Jiaru. Improved successive cancellation decoding of polar codes [J]. IEEE Transaction on Communications. 2013, 61(8): 3100–3107. DOI: 10.1109/TCOMM.2013.070213.120789.
- [7] I. TAL, A. VARDY. List decoding of polar codes [J]. IEEE Transactions on Information Theory, 2011, 61(5): 2213–2226. DOI: 10.1109/TIT.2015.2410251.
- [8] LI Bin, SHEN Hui, 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.
- [9] A BALATSOUKAS-STIMMING, A RAYMOND W GROSS, A BURG. Hardware architecture for list successive cancellation decoding of polar codes [J]. IEEE Trans. Circuits Syst. II Express Briefs, 2014, 61(8): 609–613. DOI: 10.1109/TCSII.2014.2327336.
- [10] A. BALATSOUKAS-STIMMING, M. B. PARIZI, A. BURG. LLR-based successive cancellation list decoding of polar codes [J]. IEEE Trans. Signal Process, 2015, 63(19): 5165–5179. DOI: 10.1109/ICASSP.2014.6854333.
- [11] LI Bin, SHEN Hui, TSE D. Parallel decoders of polar codes. sep. 2013 [EB/OL]. 4 sep. 2013, Available: <http://arxiv.org/abs/1309.1026>.
- [12] LI Bin, SHEN Hui, TSE D, TONG Wen. Low-latency polar codes via hybrid decoding [J]. Proc.8th Int. Symp. Turbo Codes and Iterative Inf. Process, Aug. 2014; 2223–2227. DOI: 10.1109/ISTC.2014.6955118.
- [13] YUAN B, PHARHI K K. Low-latency successive-cancellation list decoders for polar codes with multi-bit decision [J]. IEEE Trans. Very Large Scale Integr, 2015, 23(10): 2268–2280. DOI: 10.1109/TVLSI.2014.2359793.
- [14] ZHANG Zhaoyang, ZHANG Liang, WANG Xianbin, ZHONG Caijun. A split-reduced successive cancellation list decoder for polar codes [J]. IEEE Journal on Selected Areas in Communications, 2016, 34(2): 292–302. DOI: 10.1109/JSAC.2015.2504321.
- [15] CHUNG S Y, RICHARSON T J, URBANKE R L. Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation [J] IEEE Trans. Inf. Theory, 2001, 47(2): 657–670. DOI: 10.1109/18.910580.

(编辑 苗秀芝)