

北京交通大学

硕士学位论文

极化码译码可靠性测度及应用

Reliability Measurement and Its Application for Decoding of Polar Codes

作者：熊佳慧

导师：张立军

北京交通大学

2021年5月

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：

导师签名：

签字日期： 年 月 日

签字日期： 年 月 日

学校代码：10004

密级：公开

北京交通大学

硕士学位论文

极化码译码可靠性测度及应用

Reliability Measurement and Its Application for Decoding of
Polar Codes

作者姓名：熊佳慧

学 号：18120151

导师姓名：张立军

职 称：副教授

学位类别：工学

学位级别：硕士

学科专业：通信与信息系统

研究方向：信道编码

北京交通大学

2021 年 5 月

致谢

时光荏苒，三年的求学之旅即将告一段落。回顾这三年的求学生涯，有努力有汗水，有成长有收获，当然，也免不了求学之路上的磕磕绊绊。所有经历的一切，都将成为人生的宝贵财富。在此，向所有给与我帮助的人们表示真诚的感谢。

感谢我的导师张立军副教授，本篇论文从选题到完成的每一阶段都离不开张老师的悉心教导。张老师在我每次遇到困难时都会及时的给予帮助，热心地与我探讨研究中遇到的问题，发掘新的研究切入点。张老师对待科研的赤子之心和严谨的治学态度深深地影响着我，使我终身受益。在此，由衷的感谢老师给予的指导和帮助。

感谢“极化码之父”——Erdal Arıkan 教授以及所有致力于极化码研究的学者们，他们的研究成果使我找到了研究切入点。

感谢北京交通大学网络安全研究所的各位老师，感谢他们在科研和生活中给予我的关怀和帮助。感谢实验室的小伙伴们，因为他们，科研生活才变得更加的多姿多彩。感谢我的室友崔子琦，郝爽雨，我们在一起度过了三年的学习生涯，有许多美好的回忆，祝福友谊地久天长。

另外也感谢我的家人们，谢谢他们的支持和鼓励，使我能够在学校专心完成我的学业。

最后，再次对所有帮助过我的人们表示诚挚的感谢！

摘要

极化码的核心是信道极化理论,并且已证明当码长无限长时,极化码可以达到信道容量。然而对于中短码长的极化码,由于信道极化不充分,使得译码出现错误,加之缺乏译码结果检测机制,使得错误不能够被及时发现,最终导致译码性能不理想。因此,可以通过找寻译码可靠性测度,及时发现译码错误并采取对应的措施,来进一步提高译码性能。

本文主要的研究内容包括:

(1) CRC 是一种译码可靠性测度,为了进一步改善 CRC 辅助的串行抵消列表(CRC-aided successive cancellation list, CA-SCL)算法的性能,从优化 CRC 校验的角度出发,提出了基于关键集合的部分 CRC 校验辅助 SCL 的译码算法。CRC 从监督全部信息位变成了只负责监督关键集合中的元素,其中关键集合是指极化码中位于重要位置的信息比特。由于减少了 CRC 监督的信息位数,可以提高算法中 CRC 的检错准确性,进而提高译码性能。仿真结果表明,极化码在误比特率为 10^{-6} 时,部分 CRC 校验的方案较全局 CRC 校验方案可达到约 0.5 dB 的编码增益。

(2) CRC 本身并无纠错能力,它的加入还会占用部分可靠信道,使得用来传输信息比特的信道可靠性降低,针对此问题,研究了距离参数这种译码可靠性测度,并提出了基于距离参数的单比特翻转译码算法,取代原算法中的 CRC。该算法中每次翻转一个比特后重译码将获得一条译码路径,其中翻转比特取自关键集合;设定翻转次数为 T ,则最终可获得 $T + 1$ 条译码路径。分别计算每条译码路径重编码后对应的候选码字的距离参数值,择其最小值对应的译码路径为译码结果。仿真结果表明,极化码在误比特率为 10^{-5} 时,基于距离参数的单比特翻转算法较基于 CRC 的单比特翻转算法可达到约 0.2 dB 的编码增益。

(3) 上述两种译码可靠性测度都需要额外计算,为了进一步降低复杂度,本文利用信道极化特性,提出了一种新的译码可靠性测度:冻结比特差异度 δ 。该测度在译码过程中能自动得到,无需进行额外计算,并将其应用在译码算法中,提出了基于 δ 的自适应译码算法。冻结比特差异度等于 Q 集合内译码结果不为 0 的元素数目,其中 Q 集合由精心选择的冻结比特构成,这些比特根据巴氏参数或者 scaling exponent 选择,前者可利用巴氏参数的简化算法快速获得。基于 δ ,提出了自适应 SC-SCL 译码算法。进行 SC 译码后,利用 δ 判断译码结果的可靠性,只有译码结果不可靠时,才触发 SCL 译码。仿真结果表明,在信噪比为 4 dB 时,SC-SCL 算法最多仅有 1% 的几率触发 SCL 译码,译码复杂度较纯粹的 SCL 大为减少,而误比特率与纯粹的 SCL 译码持平。

关键词: 极化码; 译码可靠性测度; CRC; 距离参数; 冻结比特差异度

ABSTRACT

The core of the polar code is the channel polarization theory, and it has been proved that the polar code can reach the channel capacity when the code length is infinite. However, for polar codes with short and medium code lengths, due to the insufficient channel polarization, errors occur in decoding, and the lack of decoding results detection mechanism makes the errors unable to be discovered in time, which ultimately leads to unsatisfactory decoding performance. Therefore, the decoding performance can be further improved by searching for a measure of decoding reliability, which can find decoding errors in time and take corresponding measures.

The main contents of this paper include:

(1) CRC is a measure of decoding reliability. In order to further improve the performance of the CRC-aided successive cancellation list (CA-SCL) algorithm, from the perspective of optimizing the CRC check, a partial CRC check-aided decoding scheme based on the critical set is proposed. CRC has changed from supervising all information bits to being only responsible for supervising the elements in the critical set, where the critical set refers to the information bits located in important positions in the polar codes. As the number of information bits for CRC supervision is reduced, the accuracy of CRC error detection in the algorithm can be improved, thereby improving the decoding performance. The simulation results show that when the bit error rate of the polar code is 10^{-6} , the partial CRC check scheme can achieve a coding gain of about 0.5 dB compared with the global CRC check scheme.

(2) CRC itself has no error correction capability, and its addition will occupy part of the reliable channel, which reduces the reliability of the channel used to transmit information bits. To solve this problem, the distance parameter is proposed and given The single-bit flipping decoding algorithm based on the distance parameter replaces the CRC in the original algorithm. In this algorithm, one bit is flipped each time and then re-decoding will obtain a decoding path, where the flipped bits are taken from the critical set. Setting the number of flips to T , and finally $T+1$ decoding paths can be obtained. Calculate the distance parameter value of the candidate codeword corresponding to each decoding path after re-encoding, and select the decoding path corresponding to the minimum value as the decoding result. The simulation results show that when the bit error rate of the polar code is 10^{-5} , the single-bit flip algorithm based on the distance

parameter can achieve a coding gain about 0.2 dB compared with the single-bit flip algorithm based on CRC.

(3) Both of the above two decoding reliability measures require additional calculations. In order to further reduce the complexity, this paper uses the channel polarization characteristics and proposes a new decoding reliability measure: the freezing bit difference δ . This measure can be automatically obtained during the decoding process without additional calculations, and is applied to the decoding algorithm. An adaptive decoding algorithm based on δ is proposed.

(4) The above two decoding reliability measures require additional calculations. In order to further reduce the complexity, this paper uses the channel polarization characteristics and proposes a new decoding reliability measure: the frozen bit difference degree δ . This measure can be automatically obtained during the decoding process without additional calculations, and is applied to the decoding algorithm--an adaptive decoding algorithm based on δ is proposed. The difference degree is equal to the number of elements in the Q set whose decoding result is not 0. The Q set is composed with carefully selected frozen bits, which are selected according to the Bhattacharyya parameter or scaling exponent. The former can be quickly obtained by the simplified algorithm of the Bhattacharyya parameter. Based on δ , an adaptive SC-SCL decoding algorithm is proposed. After SC decoding, use δ to judge the reliability of the decoding result, and only trigger SCL decoding when the decoding result is unreliable. The simulation results show that when the signal-to-noise ratio is 4 dB, the SC-SCL algorithm has a 1% chance to trigger SCL decoding at most. The decoding complexity of SC-SCL algorithm is greatly reduced compared with pure SCL algorithm, but both have the same bit error rate performance.

KEYWORDS: polar codes; measure of decoding reliability; CRC; distance parameter; frozen bit difference degree

目录

摘要	iii
ABSTRACT.....	iv
1 引言	1
1.1 研究背景及意义	1
1.2 极化码的研究现状	2
1.3 创新点及结构安排	6
2 极化码的理论基础	9
2.1 信道极化原理	9
2.1.1 信道联合	10
2.1.2 信道分裂	11
2.2 极化码译码算法	12
2.2.1 SC 译码算法.....	12
2.2.2 SCL 译码算法.....	14
2.2.3 CA-SCL 译码算法	15
2.2.4 单比特翻转译码算法	16
2.3 本章小结	17
3 译码可靠性测度：CRC	18
3.1 CRC 校验的原理	18
3.2 CRC 在极化码译码中的应用	19
3.3 CS-PCA-SCL 译码算法.....	20
3.3.1 关键集合 (CS)	20
3.3.2 CS-PCA-SCL 译码算法描述.....	22
3.3.3 仿真结果及分析	23
3.4 本章小结	25
4 译码可靠性测度：距离参数	27
4.1 距离参数	27
4.2 <i>D</i> -SCF 译码算法	28
4.2.1 Chase 算法的介绍.....	28

4.2.2	<i>D</i> -SCF 译码算法描述	30
4.3.3	仿真结果及分析	32
4.3	本章小结	35
5	译码可靠性测度：冻结比特差异度	36
5.1	定义	36
5.2	特殊集合 Q 的选择策略	37
5.2.1	巴氏参数选择法	37
5.2.2	计算巴氏参数的简便算法	38
5.2.3	巴氏参数选择法的效果	43
5.2.4	scaling exponent 选择法	44
5.2.5	scaling exponent 选择法的效果	45
5.2.6	两种选择算法的对比	46
5.3	SC-SCL 自适应算法	46
5.3.1	算法描述	46
5.3.2	仿真结果及分析	47
5.4	本章小结	51
6	结论	52
	参考文献	54
	作者简历及攻读学位期间取得的研究成果	58
	独创性声明	59
	学位论文数据集	60

1 引言

1.1 研究背景及意义

随着无线通信技术的进步和迅猛发展,数字信号逐步替代了模拟信号在无线信息传输中的位置,成为一种主流的信号传输方式。但由于信道特性复杂,信号在信道传输时可能会受到干扰,比如乘性干扰,加性噪声干扰,多径效应等,这些干扰都会使得接收端在接收信号时发生错误的几率大大增加,系统的误码率也随之增大。尽管均衡的办法可以纠正一些由乘性干扰引起的误码,但对于加性噪声等造成的误码,均衡的办法就无能为力了。所以在一个实用的无线通讯系统中,必须采取一定措施来修正数据传送中的错误,提高其系统工作的可靠度。在此要求下,信道编码应运而生。信道编码,也可以称为差错控制编码,是通过在发送端以一种可控制的形式在数据信号中添加冗余比特,接收端利用添加的冗余比特来达到纠错检测的目标,从而大大提高系统的稳定和可靠度^[1]。

1948年,有噪信道编码定理被首次提出:当信号的传输速率 R 小于等于信道的容量 C 时,总是可以找到合适的信道编码技术,实现信息的高度可靠传输,使得系统的错误概率 P_e 达到任意小^[2]。 C 也被称之为香农极限 (Shannon limit)。

香农提出的信道编码理论奠定了信道编码研究的基础,香农极限也成为了信道编码领域研究者们想要达到的目标。为了逼近香农极限,许多专家学者在信道编码方向刻苦钻研,涌现出了许多行之有效的信道编码方案,比如 Hamming 码^[3], Reed-Muller 码^[4], BCH 码^[5], 卷积码^[6], Turbo 码^[7-8], LDPC 码^[9-10]等等。其中 Turbo 码和 LDPC 码性能较为优异,尤其是 LDPC 码,有学者证明当码长趋于无限时,LDPC 码理论上的译码性能离香农极限只有 0.0045 dB 的距离^[11]。由于这两种码字的优异性能, Turbo 码被确定为商用 3G 移动通信系统的关键技术,而 LDPC 码被确定为第四代通信系统的关键技术,被广泛应用于卫星数字通信,光纤通信等领域。Turbo 码和 LDPC 码的出现使信道编码研究迈入了一个新的阶段。

随着对 Turbo 码和 LDPC 码的研究的深入,很多学者都认为达到香农极限的目标近在咫尺,但是研究者却遇到一个棘手的问题,难以突破:要想达到香农极限,码长必须无限长,意味着编码必须随机。这就使得编译码复杂度大大的增加,而且对硬件要求也非常高,实现困难。直到极化码的问世,打破了这个僵局,一下引起了编码界的广泛关注。

2008年, Erdal Arıkan 教授在国际信息论 ISIT 会议上首次提出了信道极化的

概念^[12]；2009年在“IEEE Transaction on Information Theory”学术期刊上发表了一篇长达23页的论文，在这篇论文中，Arikan教授深入阐述了信道极化原理，并给出了一种新的基于信道极化的信道编码方式——极化码(polar code)^[13]。极化码具有良好的构造特性，是目前为止已知的唯一能够被严密数学验证可以达到信道容量的信道编码方案。

与其他编码不同，极化码的结构化特征更强，具有规则的、复杂度较低的编码和译码结构，并且具有良好的渐进性能和纠错性能，没有误码平层现象。极化码拥有更加灵活的速率适配方案，在实际应用中更加具有弹性和延伸性。在2016年的11月召开的3GPP RAN187大会上，由华为公司发布并主推的极化码编码方案击败了Turbo码和LDPC码，被认定是5G移动通信技术增强移动宽带(enhanced mobile broadband, eMBB)应用场景下控制信道的编码方案^[14]。极化码的成功代表着中国公司在通信领域的影响力，是里程碑式的成就。

5G是第五代移动通讯系统的简称，是4G的升级。5G的应用更加多样化^[15]，比如VR/AR，车联网，自动驾驶技术等等。5G的特点是高速度、低功耗、低时延，相比4G速率要快出一个量级左右。5G对物联网做了优化，使其具有更低的时延，能够满足虚拟现实，智能交通，通信，物联网等的需求。因此，5G成为国家提升科技和经济竞争力的重要战略方针。而极化码作为5G物理层的核心技术，将对5G系统的各项性能指标起着直接或间接的作用，重要性自然不言而喻。

虽然极化码有许多独特的优势，但毕竟历史不长，这个“后起之秀”在实际应用中仍然存在许多不足。比如，在信道极化理论的支持下，可以证明当码长无限长时，极化码在串行抵消(successive cancellation, SC)译码算法下可以达到信道容量。但对于有限长度的极化码，由于信道极化不充分，导致SC译码算法性能并未达到理想结果。而且极化码译码算法存在一个较大的问题是直接译码得到信息比特，也就无法通过码字与校验矩阵的关系来判断译码结果是否正确，不能够及时发现译码错误从而采取措施，这也是导致译码性能不理想的一个重要因素。所以为了满足实际通信场景下的可靠性要求，可以从译码可靠性测度这个角度研究有限长度下的高效极化码译码算法。

1.2 极化码的研究现状

极化码是一种较为年轻的线性分组码，在2009年由阿利坎教授提出。极化码在设计之初是一种针对二元对称信道(binary discrete symmetric channel, BSC)的严格构造码，并且已证明当码长无限长时，可以达到BSC的信道容量。

极化码的构造编码原理对信息论有很大的理论意义，为码的设计指出了努力的方向。极化码基于信道极化原理，而信道极化主要由两个阶段构成，其一是信道联合（channel combining），其二是信道分裂（channel splitting）^[16]。信道联合过程是指将多个独立信道合并成一个信道，但是信道的总容量维持不变，该过程是在编码中完成的；而信道分裂是指把接收信息 $y_1^N = (y_1, y_2, y_3, \dots, y_N)$ 分解成多个输入比特信息 $u_1^N = (u_1, u_2, u_3, \dots, u_N)$ ，是在译码的过程中完成的。经过信道联合和信道分裂之后，信道就发生了极化，具体含义主要是指虚拟子信道的对称容量呈现出了两极分化的趋势：一部分虚拟子信道的容量不断地增加，而另一部分子信道的容量将不断地减少。这种极化现象随着码长变长而更加明显，在码长无限长这种极端情况下，有部分信道的信道容量将到达容量上界 1，变为无噪声信道；而另一部分信道的信道容量将减少至容量下界 0，变为纯噪声信道。极化码的构造思想便是利用信道极化的本质，将需要传输信息的比特放在“可靠信道”中进行传输，而将不承载信息的比特——冻结比特（frozen bits）放在“不可靠信道”中传输。当码长无限长时，这种传输方法将使得传输速率达到信道容量，也就是达到香农极限。信道极化是一种普遍存在的现象，不仅存在于 BSC 信道，在加性高斯白噪声（additive white Gaussian noise, AWGN）信道也广泛存在，且码长越长，信道极化现象更加明显。

对于极化码的研究大致可以分为三个方向：信道可靠性估计，编码原理和译码算法研究。

（1）信道可靠性估计

当码长为有限长时，信道极化不完全，导致虚拟信道的信道容量大多处于 0~1 之间，也就意味着，大多数信道中仍然存在噪声干扰，这会使得接收码字可能出现错误。为了减少信息在信道传输过程中受到的干扰，需要从虚拟信道中尽可能地挑选可靠信道用来传输信息比特，挑选不可靠信道用来传输冻结比特。因此，对于极化码而言，对极化信道进行可靠性估计非常重要，将直接影响到极化码的性能表现。

极化信道的可靠性一般由三种指标来衡量，分别是对称容量 $I(W_N^{(i)})$ 、巴氏参数 $Z(W_N^{(i)})$ 和最大似然判决错误率 $P_e^{ML}(W_N^{(i)})$ 。在这三个参数中， $I(W_N^{(i)})$ 越大表示 $W_N^{(i)}$ 越可靠，而 $Z(W_N^{(i)})$ 和 $P_e^{ML}(W_N^{(i)})$ 则是越小表示 $W_N^{(i)}$ 越可靠。从目前的研究成果来看，一般不直接计算 $I(W_N^{(i)})$ 的值，因为计算公式较为复杂，并且在极化过程中， $I(W_N^{(2i-1)})$ 和 $I(W_N^{(2i)})$ 的值也没有用 $I(W_N^{(i)})$ 表示的解析的界，不方便计算。而针对 $Z(W_N^{(i)})$ 和 $P_e^{ML}(W_N^{(i)})$ 的计算更为直接，因此使用这两个参数衡量信道可靠性的研究成果更为突出。

巴氏参数法是最简单的衡量分裂子信道可靠性的方法，也是 Arikan 教授最开始在论文中提出来的度量方法。巴氏参数在物理意义上指的是信息经信道传输后

进行最大似然判决错误概率的上限,所以巴氏参数的取值范围是 $[0,1]$ 。对于巴氏参数而言,值越小表明信道越可靠。对于二进制删除信道(binary erasure channel, BEC),巴氏参数是极佳的一种度量信道可靠性的方式。在 BEC 信道下,极化虚拟信道和原始信道之间具有明确的等式关系,我们可以通过递归计算获知最终虚拟信道的巴氏参数值,从而判断信道的可靠性。但巴氏参数法的弊端在于,在其他信道条件下,极化虚拟信道和原始信道之间的巴氏参数计算公式是一个不等式,也就是说,通过递归计算只能得到巴氏参数的近似值,所以对信道可靠性的估计就不那么准确了^[17]。

因此,为了在实际应用中更加准确地判断虚拟信道的可靠性,研究者们从最大似然判决错误率出发,提出了各种估算虚拟信道错误率的算法。Mori 等人把 LDPC 码中的密度演变(density evolution, DE)的思想应用在极化码中^[18],密度演变是一种能够通过递归的计算,跟踪各个极化子信道的概率密度函数(probability density function, PDF)从而判断信道的可靠性的算法,这种算法可以被广泛地应用于任何一种类型的二进制离散无记忆信道(binary-input discrete memoryless channel, BDMC)。然而,该算法的缺点在于计算复杂度太高,实用性并不强。由于在普遍研究场景中传输信道模型被设定为 AWGN 信道,所以针对 AWGN 信道,有学者提出了高斯近似(Gaussian approximation, GA)算法^[19],通过用高斯分布来近似模拟密度进化中的对数似然比(likelihood rate, LLR)值的概率密度函数,从而简化计算。Vardy 和 Tal 基于信道退化和信道提升计算错误概率上下界,从而得到错误概率的估计值,该算法对于一般的对称信道都适用^[20]。中国公司华为提出了极化重量(polarization weight, PW)算法用于估计信道可靠性,虚拟信道的极化重量值越大,则该极化信道越可靠,该算法的优势在于可以复用,也就是说一旦获得长度为 N 的极化码子信道的极化重量后,长度为 $N/2, N/4 \dots$ 的极化码子信道的极化重量也随之获得了,无需重复计算^[21]。

判断虚拟信道可靠性的过程也称为码构造的过程,选择一个准确且简便的码构造算法对子信道进行可靠性判断是后续研究的基础。

(2) 编码原理

极化码的编码方式主要有几个特点:首先,极化码的编译码结构可以由其码长来确定;其次,极化码的编码过程并不考虑码的最小距离特性,是基于信道极化原理而选择的编码方案。

极化码编码的步骤可以大致总结为如下三步:首先,需要我们选择准确且简便的算法对极化信道进行可靠性的估计;然后需要进行比特混合操作,也就是根据极化信道的可靠性结果对信道进行排序,选择不可靠信道用于传输冻结比特,选择可靠信道用于传输信息比特;最后构造生成矩阵,极化码本身是一种二元线性分组码,

因而码字的构成可以通过信息比特序列与生成矩阵相乘。至此我们完成了极化码编码。

Arikan 教授最先提出的极化码编码方式是基于 2×2 矩阵 $F_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ 构造的规则化生成矩阵编码而成的, 生成矩阵表示为 $G_N = B_N F^{\otimes n}$, 其中 $F^{\otimes n}$ 表示对矩阵 F 进行 n 次克罗内克积运算, B_N 是对信道进行排序的矩阵, 它的作用是进行比特反序重排。信息位矩阵与生成矩阵相乘构成了编码码字, 即完成编码。这种构造方式能够构造出具有极化特性的生成矩阵, 是较为简单的一种生成矩阵构造方式^[22]。Arikan 在论文中还指出两个扩展方向: 一是可以改变基础生成矩阵的维度, 使码长不再局限于 2^n ; 二是可以扩展多进制信道, 以支持极化码更广泛地应用。

生成矩阵具有极化特性的判断依据为: 矩阵可逆, 且经过多次列变换之后主对角线全为 1, 但不构成上三角矩阵^[23]。极化性能较好的二进制小维度 ($2 \leq l \leq 16$) 极化矩阵可以通过码字分解算法来寻找^[24]。通过穷举搜索, 二进制小维度极化矩阵的线性和非线性情况被一一列举^[25]。而大维度极化矩阵的获得可以通过对小维度极化矩阵进行克罗内克积运算获得^[26], 并且通过各种降维算法, 可以改变生成矩阵的维度, 使生成矩阵维度更加灵活^[27-33]。

有研究证明多进制离散无记忆信道也存在极化现象^[34]。为了满足多进制信道极化的条件, Tanaka 和 Mori 提出用里德-所罗门码 (Reed-Solomon codes) 来构造具有极化特性的生成矩阵的算法^[35]。此外, 他们还证明使用几何码埃尔米特码 (Hermitian codes) 生成的多进制极化码具有更快的极化速率, 但多进制极化码的弊端在于生成矩阵的构造十分复杂^[36]。于是, Presman 提出混合构造极化码, 即将二进制极化码和多进制极化码混合使用^[37]。实验结果表明, 在有限码长条件下, 混合构造模式性能优于单一构造模式^[38]。

(3) 译码算法

由于在有限码长的条件下, 信道并没有达到完全极化的状态, 所以中短码长的极化码性能表现并不理想, 对于提高极化码译码性能的研究一直是研究焦点。

Arikan 教授提出的 SC 译码算法, 能够充分利用极化码的结构, 是极化码最基础的译码算法, 而且也证明了当码长趋于无限时, SC 译码算法能使极化码的性能达到信道容量。但对于中短码长, SC 译码算法的性能并不理想, 所以有许多针对极化码译码算法改进的研究。

对于译码算法的改进, 主要有两种改进思路: 其一是“嫁接”的思想。一些学者尝试将别的表现较好的译码算法“嫁接”到极化码上。比如: 置信传播 (belief propagation, BP) 算法^[39-41], 球形译码算法^[42,43], 线性规划 (linear program, LP) 算法^[44,45]等, 这些算法在一定程度上改善了译码性能, 但由于译码复杂度太高的缺点, 并未得到广泛应用。

而改进的另一个思路是针对 SC 译码算法本身的改进。Tal 和 Vardy 提出了串行抵消列表 (SCL) 译码算法^[46], 该算法通过增加候选路径的保留路径数目 L 来增大正确路径被选中的概率, 该算法是目前应用较为广泛的译码算法。在此基础上, 牛凯等人提出了一种利用循环冗余校验来辅助的串行抵消列表 (CA-SCL) 算法^[47], 进一步增大了选中候选路径中的正确路径的概率, 提升性能^[48-50]。并且他们通过在译码树模型中引入深度优先搜索的思想, 提出了基于栈结构的串行消除堆栈 (successive cancellation stack, SCS) 译码算法^[51], 紧接着将深度优先搜索和广度优先搜索进一步结合, 由此提出混合串行消除 (successive cancellation hybrid, SCH) 译码算法^[52], 这种算法充分考虑了时间复杂性和空间复杂性。由于 SC 是一种串行译码, 存在错误扩散现象。针对这一特性, 有学者提出单比特翻转的译码算法^[53], 通过每次翻转最有可能出错的比特来达到正确译码的目的。

以上就是极化码研究的三个主要方向。极化码构造思想中最重要的一条就是利用信道极化, 达到了“扬长避短”的效果, 即在较可靠的信道中传输信息比特, 在不可靠的信道中传输冻结比特, 因此性能优异。随着人们对极化码的研究不断深入, 人们对于极化码也越来越感兴趣。目前, 极化码已经应用在信息安全^[54,55], 多媒体传输^[56,57], 硬件实现^[58], 多址接入^[59], 信源编码^[60]等多个领域。

1.3 创新点及结构安排

极化码译码算法存在的一个问题是无法判断译码结果的可靠性, 这也是导致译码性能不理想的一个重要因素。本文的主要研究目标为寻找极化码译码结果的可靠性测度, 并应用在具体译码算法中以提升译码性能。创新点主要体现在以下四个方面:

(1) CRC 是一种译码可靠性测度, 认为通过 CRC 校验的码字即为正确译码结果。本文提出了针对关键集合的部分 CRC 辅助的 SCL 译码算法, 只针对关键位置进行校验, 减少了 CRC 监督的信息位数, 从而提高 CRC 校验的精度, 更有效地筛选正确的译码路径。部分 CRC 校验辅助译码方案相较于全局 CRC 辅助的 SCL 译码算法有较大的性能改善;

(2) 提出了基于距离参数的单比特翻转译码算法。距离参数是一种译码可靠性测度, 它反映的是译码码字与接收序列之间的距离。距离参数值越小, 译码路径越可靠, 由距离参数选出的译码结果也符合最大似然译码思想。新提出的算法与单比特翻转译码算法相比最大的区别在于用距离参数替代原来译码算法中的 CRC 校验以确定最终译码输出, 消除了 CRC 对译码的影响, 在一定程度上提升了译码性

能:

(3) 基于信道极化状态, 提出了另外一种译码可靠性测度: 冻结比特差异度 (δ)。 δ 通过统计属于特定范围 (Q 集合) 内的冻结比特译码硬判决结果和冻结比特设定值不一致的个数得出。若 δ 小于某一个门限值, 则认为译码正确。据此, 提出了基于冻结比特差异度的自适应 SC-SCL 译码算法, 该算法译码性能与 SCL 算法相当, 但译码复杂度大大降低。

(4) 在计算 δ 时, 需要确定 Q 集合范围。针对 Q 集合的选择策略之一——巴氏参数, 提出了计算巴氏参数的快速算法。从巴氏参数的计算公式进行研究, 发现巴氏参数具有互补性, 利用巴氏参数的这个特性, 可以很快地得到极化后虚拟子信道的巴氏参数值, 大大减少了巴氏参数的计算复杂度, 便于得到 δ 的值。

本文的结构安排如下:

第一章对研究背景及极化码的研究现状进行概述, 介绍目前极化码研究取得的主要成果;

第二章介绍了极化原理及极化码的经典译码算法。极化码是基于信道极化的一种信道编码方式, 并且已证明在码长无限长时可以达到信道容量。对于极化码而言, 最合适的译码算法就是基于 SC 译码的, 只有这类译码算法能够很好地利用极化码的结构, 并且确保在码长足够长的情况下容量可达。但中短码长下, 信道极化不充分, 导致 SC 性能表现不够理想。许多学者基于 SC 提出了改进译码算法, 比如 SCL, CA-SCL, 单比特翻转译码算法等, 这些译码算法在第二章中将会被详细介绍。

第三章讨论了 CRC 这种译码可靠性测度。CRC 校验是一种比较常见的检验译码错误的方式, 在极化码译码算法中的单比特翻转译码算法和 CA-SCL 译码算法都有用到 CRC 校验。为了进一步提高 SCL 算法性能, 本文提出了基于关键集合的部分 CRC 辅助的 SCL 译码算法, 由于关键集合的个数少于信息位个数, CRC 监督的信息个数减少, 从而提高了 CRC 校验的准确度。该算法相较全局 CRC 辅助的 SCL 译码算法性能表现更为优异。

第四章讨论了基于距离的译码可靠性测度——距离参数。距离参数是通过衡量译码码字与接收序列间的距离而得到的, 距离参数值越小, 译码结果越可靠, 这也符合最大似然译码准则。此外提出了基于距离参数的单比特翻转译码算法, 该算法中用距离参数取代了 CRC 进行译码结果的衡量并选择出正确译码路径, 消除了 CRC 的影响, 进一步提升性能。

第五章提出了另一种译码可靠性测度: 冻结比特差异度 (δ)。该参数是指特定范围 (Q 集合) 内的冻结比特译码硬判决结果和冻结比特设定值不一致的个数, 针对 Q 集合的选择, 给出了两种选择算法: 巴氏参数法和 scaling exponent 选择法。

当 δ 的值小于设定的门限值时,则认为译码结果正确。将 δ 应用在实际译码算法中,提出了 SC-SCL 自适应译码算法,该算法译码性能与 SCL 持平,但复杂度大大降低。

第六章总结全文并对未来的研究做出展望。

2 极化码的理论基础

极化码是一种比较新的线性分组码，基础原理是信道极化。目前唯一在理论上被证明可以在码长为无限长的条件下达到 BSC 的信道容量的编码方式就是极化码。为了便于对后文的理解，本章将对极化码的极化原理和典型译码算法进行阐述。

2.1 信道极化原理

一个二进制的离散无记忆信道简单的表示形式为 $W: X \rightarrow Y$ ，其中 X 表示信道的输入符号集合，集合 $X = \{0,1\}$ ； Y 表示信道的输出符号集合，则该信道的转移概率可以写成 $W(y|x), x \in X, y \in Y$ 。 Y 和 $W(y|x)$ 是任意值。定义对信道 W 进行 N 次使用后的信道表示为 W^N ，则信道 $W^N: X^N \rightarrow Y^N$ 的转移概率可以简单的表示成公式(2-1)：

$$W^N(y_1^N|x_1^N) = \prod_{i=1}^N W(y_i|x_i) \quad (2-1)$$

对于信道 W 而言，有两个信道参数非常重要，分别是对称容量 $I(W)$ 和巴氏参数 $Z(W)$ ^[13]。计算公式如 (2-2) 和 (2-3)：

$$I(W) \triangleq \sum_{y \in Y} \sum_{x \in X} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)} \quad (2-2)$$

$$Z(W) \triangleq \sum_{y \in Y} \sqrt{W(y|0)W(y|1)} \quad (2-3)$$

其中对称容量 $I(W)$ 的物理意义是指信道 W 在一个等概的输入情况下可靠传输时的最大速率，可以用来对信道速率进行度量；而 $Z(W)$ 的物理意义是信道 W 只传输 0 或 1 时的最大似然判决错误概率的上限，由此看出， $Z(W)$ 是可以用来度量信道的传输可靠性的。关于巴氏参数还会在第五章进行详细介绍。

极化码的基本原理是信道极化定理。而信道极化主要由两个阶段构成，其一是信道联合，其二是信道分裂。当信道极化之后，总信道的容量仍然保持不变，但各虚拟信道的信道容量发生了变化，有一部分信道的信道容量会随之增加，成为更可靠的信道；而另一部分信道的信道容量会随之减少，成为不可靠的信道。这种极化现象随码长增加而变得更加地明显。码长趋于无限时，可靠信道的信道容量将趋近于容量上限 1，成为无噪声的完美信道，而不可靠信道的信道容量将趋近容量下限 0，成为纯噪声信道。

2.1.1 信道联合

信道联合,是指将同一个二进制离散无记忆信道 W 采用复用的方法反复使用 N 次,最终形成一个合成信道 W_N 的过程: $W^N: X^N \rightarrow Y^N$, 其中 N 为 2 的幂,即 $N = 2^n, n \geq 0$ 。举一个最简单的例子,当 $N = 2$ 时,信道联合过程如图 2-1 所示,得到向量信道 $W_2: X^2 \rightarrow Y^2$ 。

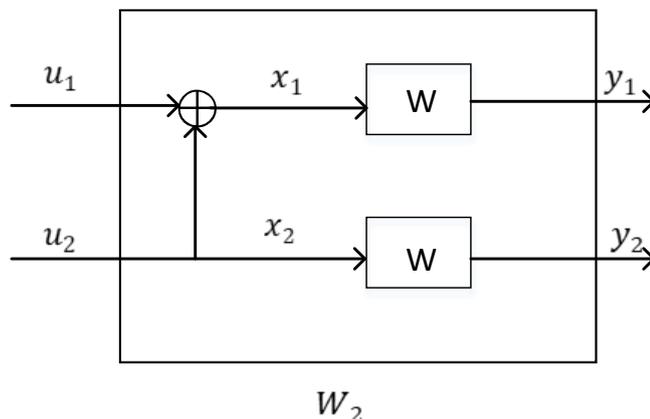


图 2-1 合并信道 W_2 ^[13]
Figure 2-1 Combining channel W_2 ^[13]

其对应的转移概率如公式 (2-4):

$$W_2(y_1, y_2 | u_1, u_2) = W(y_1 | u_1 \oplus u_2) W(y_2 | u_2) \quad (2-4)$$

同理,当码长 $N = 4$ 时,信道联合过程如图 2-2 所示,通过联合信道 W_2 的 2 个独立副本得到信道 $W_4: X^4 \rightarrow Y^4$ 。

该过程的转移概率如公式 (2-5) 所示:

$$W_4(y_1^4 | u_1^4) = W_2(y_1^2 | u_1 \oplus u_2, u_3 \oplus u_4) W_2(y_3^2 | u_2, u_4) \quad (2-5)$$

在图 2-2 中, R_4 是置换操作矩阵。若输入比特序列为 (s_1, s_2, s_3, s_4) , 则经过置换操作后,得到比特序列 $v_1^4 = (s_1, s_3, s_2, s_4)$ 。经过信道 W_4 , $u_1^4 \rightarrow x_1^4$ 的映射可用公式表示为

$$x_1^4 = u_1^4 G_4 \quad (2-6)$$

因此 W_4 和 W^4 的转移概率如公式 (2-7)

$$W_4(y_1^4 | u_1^4) = W^4(y_1^4 | u_1^4 G_4) \quad (2-7)$$

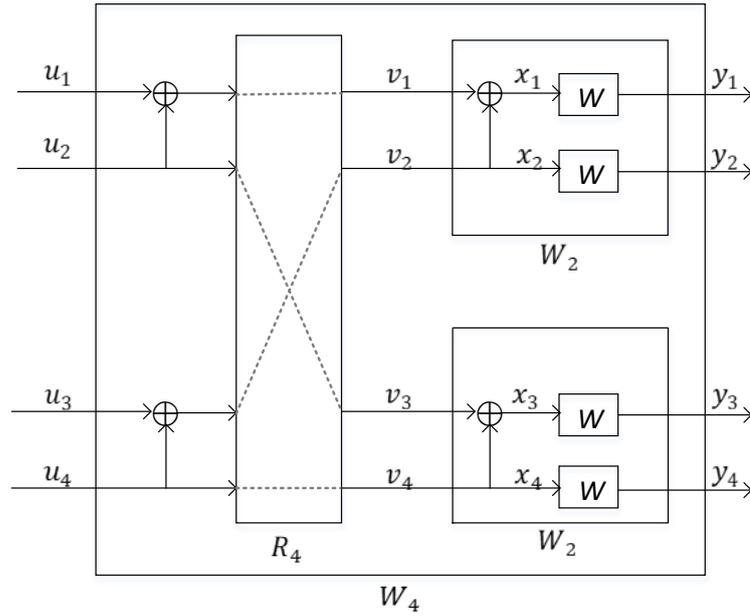


图 2-2 合并信道 W_4 ^[13]
Figure 2-2 Combining channel W_4 ^[13]

由此类推，经过信道 W_N ， $u_1^N \rightarrow x_1^N$ 的映射可用公式表示为

$$x_1^N = u_1^N G_N \tag{2-8}$$

称 G_N 为 N 维生成矩阵。所以信道 W_N 和 W^N 的转移概率如公式（2-9）：

$$W_N(y_1^N | u_1^N) = W^N(y_1^N | u_1^N G_N) \tag{2-9}$$

其中 $y_1^N \in Y^N, u_1^N \in X^N$ 。需要注意的是，信道联合不会改变信道总容量，也就意味着各个虚拟子信道的容量之和加起来等于合并信道的总容量，即满足容量守恒。

2.1.2 信道分裂

信道分裂，是指将合并信道 W_N 重新拆分成 N 个虚拟子信道，但是这些虚拟子信道之间不再彼此独立，而是有一定的联系，这是信道极化的第二个阶段。其中第 i 个虚拟信道 $W_N^{(i)}: X \rightarrow Y^N \times X^{i-1}, 1 \leq i \leq N$ ，定义其转移概率如公式(2-10)所示：

$$W_N^{(i)}(y_1^N, u_1^{i-1} | u_i) \triangleq \sum_{u_{i+1}^N \in X^{N-i}} \frac{1}{2^{N-i}} W_N(y_1^N | u_1^N) \tag{2-10}$$

其中 (y_1^N, u_1^{i-1}) 表示 $W_N^{(i)}$ 的输出，而 u_i 表示 $W_N^{(i)}$ 的输入。

注意奇序子信道和偶序子信道的转移概率公式不同：

$$W_N^{(2i-1)}(y_1^N, u_1^{2i-2} | u_{2i-1}) = \sum_{u_{2i}} \frac{1}{2} W_{\frac{N}{2}}^{(i)} \left(y_1^{\frac{N}{2}}, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2} \middle| u_{2i-1} \oplus u_{2i} \right) \cdot W_{\frac{N}{2}}^{(i)} \left(y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2} \middle| u_{2i} \right) \quad (2-11)$$

$$W_N^{(2i)}(y_1^N, u_1^{2i-1} | u_{2i}) = \frac{1}{2} W_{\frac{N}{2}}^{(i)} \left(y_1^{\frac{N}{2}}, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2} \middle| u_{2i-1} \oplus u_{2i} \right) \cdot W_{\frac{N}{2}}^{(i)} \left(y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2} \middle| u_{2i} \right) \quad (2-12)$$

其中 $n \geq 0, N = 2^n, 1 \leq i \leq N/2$ 。

信道分裂的过程实际上就是 SC 解码过程。需要注意的是，信道分裂过程也是容量守恒的。

2.2 极化码译码算法

从理论上说，码长无限长时极化码在 SC 译码算法下是可以达到信道容量的。但是在实际应用中，码长不可能达到无限长，这会导致信道极化并不那么充分，无法得到理想的译码性能。所以寻找低复杂度的，性能优异的译码算法对于极化码的推广至关重要。SC 译码算法是极化码最基础的译码算法，除 SC 译码算法之外，也有许多学者研究其他译码算法在极化码中的应用，比如 BP 译码算法，球形译码算法等，但这些算法性能提升有限且复杂度较高，实用性受到限制；所以针对译码算法的改进更普遍的一种研究思路是基于 SC 译码算法的改进。本小节将介绍 SC 译码算法及典型的 SC 改进译码算法。

2.2.1 SC 译码算法

从第一章介绍的极化码编码原理可知，极化码的构造和编码过程其实就是信道的选择过程，而信道的选择实际上与 SC 译码算法是配套的。信道经过信道联合和信道分裂后，已经不再独立，虚拟子信道之间变得相互依赖：一个序号更大的子信道会依赖于所有比它的序号小的子信道。SC 译码算法正是充分利用了虚拟子信道的这种依赖关系，在进行译码时，对当前比特进行译码判决需要假定之前所有译码全部正确。对于极化码而言，最合适的译码算法应该是 SC，它充分利用了极化码本身的构造特性。阿利坎教授已经证明在 SC 译码算法下，极化码是可以达到信

道容量的。

SC 是一种串行的译码算法：对信道序号从 1 到 N 取值，按照固定次序从上至下依次译码，各个比特的估计值如公式 (2-13) 所示：

$$\hat{u}_i = \begin{cases} h_i(y_1^N, \hat{u}_1^{i-1}), & \text{if } i \in A \\ u_i, & \text{if } i \in A^c \end{cases} \quad (2-13)$$

其中，当 $i \in A^c$ 时，表明该比特为冻结比特，即收发双方已经提前约定好的比特，此时可以直接进行判决；当 $i \in A$ 时，表明该比特为承载信息的比特，需要依据对数似然比的值进行判决，判决函数如公式 (2-14) 所示：

$$h_i(y_1^N, \hat{u}_1^{i-1}) = \begin{cases} 0, & \text{if } L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 0 \\ 1, & \text{if } L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) < 0 \end{cases} \quad (2-14)$$

公式 (2-14) 中定义的对数似然比 $L_N^{(i)}$ 为

$$L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \triangleq \ln \left(\frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)} \right) \quad (2-15)$$

LLR 的计算可以通过递归计算，奇序分裂子信道的 LLR 值计算如公式 (2-16)，偶序分裂子信道的 LLR 值计算如公式 (2-17)：

$$L_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}) = f \left(L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}), L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}) \right) \quad (2-16)$$

$$L_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = g \left(L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}), L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{2i-1}^{2i-1}) \right) \quad (2-17)$$

公式 (2-16) (2-17) 之中的函数 f 和 g 定义为

$$f(a, b) = \ln \left(\frac{1+e^{a+b}}{e^a+e^b} \right) \approx \text{sign}(a)\text{sign}(b)\min\{|a|, |b|\} \quad (2-18)$$

$$g(a, b, u_s) = (-1)^{u_s}a + b \quad (2-19)$$

至此，可以通过译码端收到的似然比值进行递归计算，直到到达了信道 W 端，通过判决公式 (2-13) 得到硬判决结果，即为译码输出。

SC 译码算法是通过递归计算得到虚拟子信道的 LLR 值，并根据 LLR 值进行硬判决得到最终的译码结果，而且 SC 译码算法是按照严格的顺序从上往下依次译码。当码长无限长的时候，信道已经完全极化，极化码按照 SC 译码算法在理论上可以达到信道容量。SC 译码器的复杂度仅为 $O(N \log N)$ ，复杂度较低。可以说 SC 译码算法是极化码最核心的译码方式。但 SC 译码算法的问题在于：当信道极化不完全时，会存在一些信息比特无法被正确译码，但由于 SC 算法存在错误传递的问题，前面比特的错误判决将会向后传递，从而影响后续的译码。因此，对于有限码

长的极化码，采用 SC 译码器往往不能达到理想的性能。

2.2.2 SCL 译码算法

SCL 译码算法是一种较为成功的 SC 改进译码算法，大大提高了译码的性能，它也是当前应用较为广泛的极化码译码算法。

针对 SC 译码的错误传播特性，国外学者提出了一个直接的改进方案，命名为 SCL 译码算法。SCL 译码算法主要是通过对每一个比特进行路径扩展，在每一层的路径搜索中都允许保留 L 条候选路径，这样一来，会大大增加正确路径被保留的概率。若在某一层的路径数超过 L 时，需要对候选路径列表进行删减，通过评估候选路径的路径度量值（path metrics, PM），保留 PM 值最小的 L 条路径，然后继续向下一层进行路径的扩展。通过多保留候选路径这种方式，有效避免了错误传播现象。一般将参数 L 称为搜索宽度，SCL 译码算法相当于 L 个 SC 译码器同步地进行，当 $L=1$ 时，SCL 译码算法即为 SC 译码算法。

SCL 译码算法流程总结如下：

1) 初始化。将搜索宽度 L 以及路径度量 PM 初始化为 0，记 $L^{(0)} = \{\emptyset\}$, $PM(\emptyset) = 0$ 。

2) 路径扩展。第 i 个信息比特译码时，将前 $i-1$ 层的候选路径保存在列表内，然后对第 i 层每个节点进行 0 或 1 的路径扩展，将扩展得到的所有候选路径保存在列表 $L^{(i)}$ 中，更新 $L^{(i)}$ 中所有候选路径的度量值 $PM(u_1^i)$ ，其中 $PM(u_1^i) = -\ln(\Pr\{u_1^i|y_1^N\})$ 。

3) 路径竞争。若候选路径数 $|L^{(i)}| > L$ ，则将列表 $L^{(i)}$ 中保存的 PM 值升序排序，选出前 L 条 PM 值较小的候选路径用于下次路径扩展；若候选路径数 $|L^{(i)}| \leq L$ ，则保留所有候选路径用于下次路径扩展。

4) 判决结束。若译码序号 $i < N$ ，则需要继续执行步骤 2；若 $i = N$ ，则从路径列表 $L^{(N)}$ 中筛选出 PM 值最小的一条路径，输出其对应的译码序列，结束译码。

在文献[62]给出了路径度量 PM 的定义：

$$-\ln\Pr(u_1^i|y_1^N) = \sum_{k=1}^i \ln(1 + e^{-(1-2u_k)L_N^{(k)}}) \quad (2-20)$$

从 PM 的定义可以看出，PM 的值越小越好。若 $u_k = 0, L_N^{(k)} \geq 0$ ，则 $PM \approx 0$ ；若 $u_k = 1, L_N^{(k)} \geq 0$ ，则 $PM \approx L_N^{(k)}$ ；若 $u_k = 0, L_N^{(k)} \leq 0$ ，则 $PM \approx -L_N^{(k)}$ ；若 $u_k = 1, L_N^{(k)} \leq 0$ ，则 $PM \approx 0$ 。从这四种情况可见，当 u_k 的值符合 $L_N^{(k)}$ 的硬判决时，PM 的值总是约为 0；反之，当 u_k 的值不等于 $L_N^{(k)}$ 的硬判决时，PM 的值总是约等于 $|L_N^{(k)}|$ 。所以， u_k 的值不等于 $L_N^{(k)}$ 的硬判决将会导致 PM 值增大 $|L_N^{(k)}|$ ，可以理解为是一个惩罚值。

SCL 译码算法的问题在于会增大译码开销，译码复杂度增加。于是有一篇论文针对 SCL 译码算法提出一种懒拷贝技术，使得 SCL 译码算法的空间复杂度下降为 $O(LN)$ ，时间复杂度降为 $O(LN\log N)$ ，从而减少极化码在硬件上对资源的占用，从而拓宽极化码的应用范围^[46]。

2.2.3 CA-SCL 译码算法

为了进一步增大 SCL 译码算法中选中正确路径的概率，有学者提出了 CA-SCL 译码算法^[47]。CRC 是一种信道检错技术，它在实践中被广泛地应用在数字通信系统。对于极化码而言，在 SCL 译码算法中，我们可以使用 CRC 对最终的 L 条候选路径做出正确的选择，利用“正确路径可以通过 CRC 校验”的先验信息，增大选中正确译码路径的概率，从而提升译码性能。

CA-SCL 译码过程如图 2-3 所示：

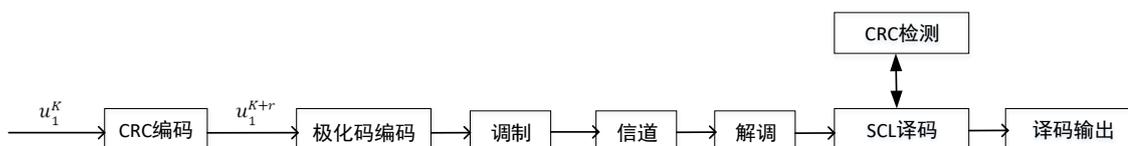


图 2-3 CA-SCL 译码算法流程框图
Figure 2-3 CA-SCL decoding algorithm flow diagram

在进行极化码编码之前，通过在信息比特序列后添加 CRC 校验比特构成新序列，新序列作为新的信息序列进行极化码编码后送入信道传输。利用 SCL 译码算法正常译码获得 L 条译码路径，然后借助“正确译码结果可以通过 CRC 校验”的先验信息，从这 L 条译码路径中选择能够通过 CRC 校验的路径作为译码结果输出，具体操作方式为：将 L 条译码路径按路径度量值从小到大的顺序依次进行 CRC 检测，一旦某条路径通过校验，则将该路径输出为译码结果，终止译码。若最终所有路径均未通过校验，则从候选路径列表中选择路径度量值最小的作为本次译码结果。

CA-SCL 译码算法的基本思想其实与 SCL 算法是一致的，只不过在 SCL 算法的基础上增加了 CRC 用于对译码结果进行校验，从而增大了选择正确译码结果的概率，提升了译码性能。

2.2.4 单比特翻转译码算法

单比特翻转译码算法是基于 SC 算法的另一种改进译码算法。由于 SC 是一种串行译码，前面译码比特的结果将会影响到后面的译码结果，所以为了防止错误传递，应当及时纠正第一个译码错误的比特。单比特翻转译码算法正是基于这样的思想，找到最有可能出错的位置并将其进行翻转，之后再重新进行 SC 译码过程。

比特翻转译码器使用的并不是单纯的极化码，而是 CRC 和极化码构成的简单级联码。CRC 在算法中的作用主要是判断翻转位置是否正确以及决定是否该终止译码。

单比特翻转译码算法流程如下：

(1) 对接收信号 y_1^N 执行传统的 SC 译码过程，得到级联码信息比特的估计 $[\hat{u}_1^K, \hat{c}_1^r]$ ；

(2) 用 CRC 校验 $[\hat{u}_1^K, \hat{c}_1^r]$ 。如果 $[\hat{u}_1^K, \hat{c}_1^r]$ 满足 CRC 校验，或者说 $[\hat{u}_1^K, \hat{c}_1^r]$ 是一个 CRC 码字，则译码结束；如果 $[\hat{u}_1^K, \hat{c}_1^r]$ 未通过 CRC 校验，则需要翻转某一信息比特后再次执行 SC 译码，这时设最大允许翻转次数为 T ，当前比特翻转次数 $t = 1$ ，转入步骤 (3)；

(3) 如果 $t > T$ ，翻转次数超过了预设值，宣布译码失败；如果 $t \leq T$ ，选出一个最不可靠信息比特 $u_i, i \in A$ ，在判决 u_1, \dots, u_{i-1} 的过程中，执行传统 SC 过程。判决 u_i 时，将 u_i 的值按照其对应的 LLR 的反方向判决：如果 $\ln \frac{w_N^{(i)}(y_1^N, u_1^{i-1}|0)}{w_N^{(i)}(y_1^N, u_1^{i-1}|1)} \geq 0$ ，则 $u_i = 1$ ，反之 $u_i = 0$ 。随后，判决 u_{i+1}, \dots, u_N 的过程中，继续执行传统 SC 过程。

(4) 如果本次译码结果通过了 CRC 校验，则译码结束；如果仍未通过 CRC 校验，则 $t = t + 1$ ，返回第 (3) 步选另外一个 u_i 进行翻转。

从算法描述可以看出，单比特翻转算法的关键在于翻转位置的选择，以保证该 u_i 是 SC 译码中的首个错误比特，因为必须纠正首个错误比特才有可能获得正确的译码。有文献提出可以选择 u_s ，其中

$$s = \arg \min \left\{ \left| \ln \frac{w_N^{(i)}(y_1^N, u_1^{i-1}|0)}{w_N^{(i)}(y_1^N, u_1^{i-1}|1)} \right| \right\} \quad i \in A \quad (2-21)$$

在随后的比特翻转译码过程中翻转 u_s 的值即可，因为小的 LLR 绝对值表示判决不可靠^[53]。有学者通过分析极化码的结构，提出了关键集合 (critical set, CS) 的概念^[61]，认为关键集合是第一个译错的比特位置的集合，如此，便只需翻转关键集合中的元素即可。关于关键集合的知识，在第三章将会进行详细的介绍。

研究证明，单比特翻转译码算法的译码性能要远远优于单纯的 SC 译码算法，但复杂度不能确定，也许只翻转一次，译码结果便可通过 CRC 校验，成功译码，也有可能达到最大翻转次数也未能成功译码。

2.3 本章小结

本章主要介绍了极化码的一些基本概念和极化码的主要译码算法。

极化码源自信道极化现象，包括信道联合和信道分裂。信道极化过后得到的虚拟子信道相互依赖。虽然信道总容量不变，但信道容量会呈现两极分化的趋势，一些子信道的容量会增大，而另一些会减少，若码长足够长，最终子信道的信道容量将为 1（无噪声信道）或 0（纯噪声信道）。

极化码最有效的译码算法是 SC 译码算法，该算法充分利用了极化码的结构特征，但由于在中短码长下，信道极化并不完全，导致 SC 算法的性能并不理想。在 SC 译码算法的基础上，学者们提出了很多改进性能算法，比如单比特翻译译码算法，SCL，CA-SCL 等，极大地提高了译码性能，使得极化码的应用更加广泛。

3 译码可靠性测度：CRC

信号在信道进行传输时会受到噪声的干扰，因而在接收端接收到的信息可能出现错误。受到的噪声干扰越强，译码出现错误的概率就越大。为了进一步降低误码率，可以先对译码码字可靠性进行判断，从而决定是否需要采取别的措施继续译码。对于极化码，由于其特殊的编译码结构，不能直接利用码字与校验矩阵 H 的关系来判断译码是否正确，需要额外的方法来辅助判断译码结果的正确性。CRC 校验是一种比较常见的方式。

3.1 CRC 校验的原理

一个好的校验方法，可大大提高通信系统的健壮性和稳定性。CRC 是一种比较常见的信道编码技术，一般用来检测数据经信道传输后可能出现的错误。CRC 校验位是根据数据特定生成的，然后添加于数据之后构成新的数据帧发送。由于 CRC 校验具有检错能力强，复杂度相对较低，易于硬件实现等优势，被应用在多种通信实际场景中。

将 CRC 的校验原理总结如下：首先需要根据数据生成对应的 CRC 校验位，使得加上校验位之后的新的数据帧能够被收发双方约定好的某个数整除。新的数据帧通过信道传送给相应的接收端之后，接收端需要进一步验证该数据帧的正确性：将接收端得到的数据帧除以已经约定好的除数，判断余数是否为 0。若不是 0，则可以认为数据在传输的过程中出现了错误；若为 0，则认为接收到的数据帧都是正确的，通过校验。需要注意的是，计算过程中的除法均指模 2 除法。

下面通过一个实例具体说明 CRC 校验的操作过程：

假设现有数据帧 10110011 需要传输。选定的 CRC 的生成多项式为

$$G(X) = X^4 + X^3 + 1 \quad (3-1)$$

则该生成多项式对应的二进制比特串为 11001。

(1) 由生成多项式可以确定 CRC 校验位的位数为 4 个，因此，需要在原始数据帧的后面添加 4 个 0，即 10110011 \rightarrow 101100110000。用得到的新的数据帧除以我们选定的生成多项式，即可得到余数。在此例中，最终求得余数为 0100，即 CRC 校验位也是 0100。

(2) 将 (1) 得到的 CRC 校验码拼接在原始数据帧之后，即 10110011 \rightarrow 101100110100。得到的新帧将通过信道传输给接收端。

(3) 接收端接收到数据帧后，需要判断这个数据帧是否存在错误。具体做法为：用数据帧除以约定好的生成多项式对应的二进制比特串，即 11001，判断余数值是不是为 0。若余数为 0，则说明数据帧是正确的，未发生错误；反之，则认为传输过程中受到噪声干扰，导致误码的出现。

需要注意的是，CRC 的准确度并不是 100%，有可能出现漏判的情况，但概率较低，所以 CRC 的准确度还是相当高的。想进一步提高 CRC 校验的准确性，可以通过增加 CRC 校验位数等方式。

3.2 CRC 在极化码译码中的应用

由于极化码本身在译码结束后缺乏判断译码是否成功的机制，所以需要增加额外的校验方法对极化码译码结果的正确性进行判断。CRC 校验在极化码译码算法中的应用，较为典型的算法有单比特翻转译码算法和 CRC 辅助的 SCL 译码(CA-SCL) 算法。

在单比特翻转译码算法中，CRC 的作用主要是判断翻转位置是否正确以及决定是否该终止译码。在单比特翻转译码算法中，首先确定 CRC 的生成多项式，并根据信息比特序列确定校验位，拼接在信息序列之后形成含 CRC 的简单级联码。级联码经过信道传输后，在译码端通过 SC 译码算法进行译码。判断译得的码字（包含 CRC 校验位）与已知除数的模 2 除法运算所得到的余数是否为 0，若为 0，则认为本次 SC 译码结果可靠，直接输出最终译码结果；若不为 0，则选择虚拟子信道中 LLR 值最小的位置进行硬判决结果的翻转，之后重新进行 SC 译码，译码结果再次进行 CRC 校验，以判断翻转位置是否正确以及此次译码是否成功。若经过翻转后仍未通过 CRC 校验，则继续翻转，直到达到最大翻转次数为止。

在 CA-SCL 译码算法中，CRC 校验的主要功能是在最后 L 条候选路径中选出正确的译码路径，以提升译码性能。CA-SCL 译码算法使用的也是 CRC 与极化码的简单级联码。在译码过程中，先进行 SCL 译码，以得到 L 条译码候选路径，然后再借助“正确译码结果可以通过 CRC 校验”的前提条件，从这 L 条译码路径中选择能够通过校验的路径作为译码结果的输出，具体操作为：将 L 条译码按路径度量值从小到大的顺序排序，然后依次进行 CRC 检测，一旦某一条路径通过校验，则将该路径输出为译码结果，并且停止译码。若无路径通过 CRC 校验，则从候选路径列表中选择路径度量值最小的译码路径作为本次译码结果。

CRC 校验算法嵌入在极化码的译码算法中，可以进一步改善极化码的译码性能。比如，单比特翻转译码算法与 SC 译码算法性能比较，它的 BLER 性能优于单

纯的 SC 译码器^[53]。而通过将 CRC 校验应用在 SCL 译码算法中，也大幅度提升了译码性能，尤其在高信噪比下，提升更加明显，性能更为优异。CA-SCL 译码算法是目前应用最广泛的译码算法。

3.3 CS-PCA-SCL 译码算法

想进一步提高 CRC 校验的准确性，可以通过增加 CRC 校验位数等方式，但对于极化码译码而言，这种方式是有风险的。因为 CRC 校验位数越多，拼接在信息比特序列后进行传输时，会占用可靠信道；相对应的会占用更多不可靠的虚拟子信道用于传输信息比特，反而会造成误码的出现。这就陷入了一个非常矛盾的境地：一方面可以通过增加 CRC 校验位来提升 CRC 校验的准确度，从而加大选中正确译码路径的概率；但另一方面，增大 CRC 校验位长度又会损害译码性能。实际上，对于一个特定码长，特定码率的极化码而言，存在最佳 CRC 校验位长度使得极化码性能达到最佳，但这需要进行大量的蒙特卡罗仿真实验才能够确定，效率低下。

那么有没有更加通用的办法，在不增加 CRC 校验位的条件下提高 CRC 校验的准确度，从而提升译码性能呢？针对此问题，本小节提出基于关键集合的部分 CRC 辅助的 SCL 译码 (Based on Critical Sets-Partial CRC Aided SCL, CS-PCA-SCL) 算法，即通过只对关键集合做 CRC 校验来提升 CRC 校验的准确度，进而达到提升译码性能的目的，而 CRC 校验位位数保持不变。

3.3.1 关键集合 (CS)

关键集合 (CS) 是指信息比特中的关键集合。关键集合是通过分析极化码的结构而来的，它的物理意义是指在一次译码中，译错的第一个信息比特 99% 都位于关键集合中^[61]。

寻找关键集合的算法如下：

(1) 衡量虚拟子信道的可靠性，并选出信息比特位置和冻结比特位置，根据现在极化码的码构造情况绘制极化码的树图；

(2) 将极化码划分成若干个码率为 1 的子码，则每个字块中的第一个信息比特即为关键集合中的元素。

下面给出码长 $N = 16$ ，信息位个数 $K = 9$ 的实例进一步说明关键集合的寻找过程：

1) 树图的构建

通过巴氏参数法求得各个虚拟子信道的巴氏参数值并按值从小到大进行排列，得出用于传递信息比特的信道序号为 $[u_6, u_7, u_8, u_{11}, u_{12}, u_{13}, u_{14}, u_{15}, u_{16}]$ 。则极化码的码树如图 3-1 所示。

其中，叶子节点中的白色圆圈代表冻结比特节点，黑色圆圈代表信息比特节点。对于非叶子节点，若它的两个子节点的颜色相同，则该非叶子节点保持与子节点的颜色一致，比如该节点的两个子节点均为白色圆圈，则该节点也为白色圆圈；若它的两个子节点的颜色不一致，则该非叶子节点为灰色圆圈。

2) 寻找关键集合元素

首先应找到码率为 1 的子码：从图中看，则应寻找从根节点到叶子节点均为黑色圆圈的子树即为码率为 1 的子码，如图 3-1 中的 A, B, C, D 四个子树。每个子树上对应的叶子节点中的第一个即为关键集合中的元素。此实例中，关键集合 $CS = [u_6, u_7, u_{11}, u_{13}]$ 。关键集合 CS 的元素个数恰好等于码率为 1 的子码的个数。

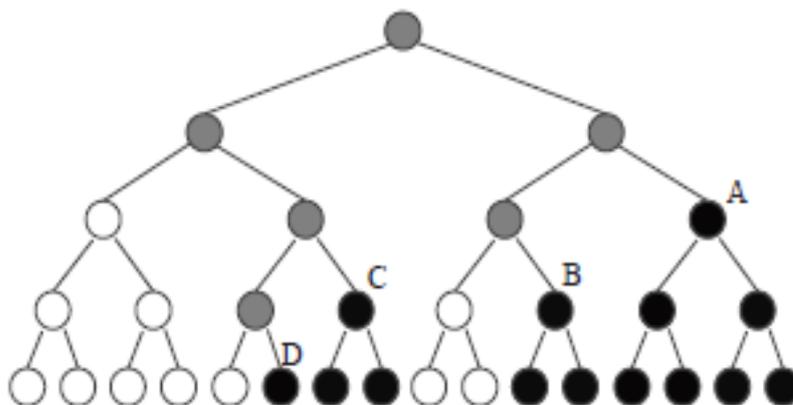


图 3-1 (16,9) 极化码树图^[61]
Figure 3-1 (16,9) polar code tree diagram^[61]

为什么由此选出来的关键集合中的元素极大概率是一次译码中第一个译错的信息比特？文献[61]给出了较为详细的证明，在此不再赘述。且文献给出蒙特卡罗仿真的数据来验证关键集合的准确性。表 3-1 给出了码长 $N = 1024$ ，信息位长度 $K = 512$ 的极化码在加性噪声高斯信道下的仿真结果，在每一个信噪比下，都仿真了 10^6 个码字。其中 N_E 表示在 10^6 次仿真中，译码出现错误的次数； N_S 表示当译码发生错误时，第一个译码错误的信息比特落在关键集合 CS 中的次数；则由 N_E 和 N_S 便可很方便地计算出关键集合用于预测错误的信息比特的准确度： $A = N_S/N_E$ 。表中还给出了在不同信噪比下的关键集合的元素个数。

表 3-1 不同信噪比下关键集合判断错误比特的准确性^[61]
 Table 3-1 The accuracy of the critical set to judge the error bit under different SNR ^[61]

	1 dB	1.5 dB	2 dB	2.5 dB	3 dB
N_E	677211	296573	73810	10888	1007
N_S	675840	296391	73789	10888	1007
A	99.8%	99.94%	99.97%	100%	100%
关键结合元素个数	110	112	117	124	129

从表格 3-1 中的数据可以观测到，即使在低信噪比下，关键集合预测第一个出错的信息比特的准确率可以达到 99%以上，而在较高信噪比下，预测准确度能达到 100%。由此可以推断，关键集合中的元素极大地影响译码性能，找出关键集合后基本上就锁定了译码出错的位置。

3.3.2 CS-PCA-SCL 译码算法描述

从上一小节中对关键集合的描述可以看出，假如一次译码出现错误，则发生译码错误的信息比特极大概率是关键集合中的元素。所以本章提出的 CS-PCA-SCL 译码算法在原 CA-SCL 译码算法的基础上基于关键集合做以下两点改进：

1) 在 SCL 译码过程中，不再针对每个信息比特都进行路径分裂，而只针对于关键集合中的信息比特进行路径分裂。

在传统 SCL 译码算法中，每译码至信息比特 u_i ，都会在此进行路径分裂，其中一个路径将 u_i 硬判决为 0；另一条路径将 u_i 硬判决为 1，分别计算两种情况下的路径度量值。 u_i 的硬判决结果将会随之向下传递，影响后续信息比特的译码。既然译错的信息比特大概率存在于关键集合中，那么就无需对所有的信息比特都进行路径分裂，而只针对处于关键集合中的信息比特进行路径分裂，对于不属于关键集合中的信息比特，则根据似然比 LLR 的值进行硬判决。这样一来，可以极大地减少运算复杂度和空间复杂度，且不会降低译码性能。

2) 不需要再针对全部信息比特进行全局的 CRC 校验，而只针对于部分信息比特——关键集合中的信息比特进行 CRC 校验。

在 CRC 校验位长度一致的情况下，CRC 校验的信息位数越少则校验越精准。由于关键集合中信息比特不可靠，较大概率出错，那么只要保证关键集合中的信息位译码不出现错误，能够通过 CRC 校验，就能大概率地保证译码正确。因此使用 CRC 校验关键集合中的元素，能够减少 CRC 监督信息位的个数，进一步提升 CRC 校验的准确度，优化译码性能。

3.3.3 仿真结果及分析

为了验证 CS-PCA-SCL 译码算法的有效性,用 MATLAB 模拟在高斯信道下的通信系统进行了仿真实验,将仿真结果总结如下:

图 3-2 表示 (128,68) 的 CRC 与极化码的级联码在 CA-SCL 译码算法与 CS-PCA-SCL 译码算法在高斯信道下的仿真结果对比,其中两种译码算法设置的 CRC 校验位长度均为 4,只不过两种算法下 CRC 监督的信息位个数不同,SCL 算法中的搜索宽度 L 均设置为 4:

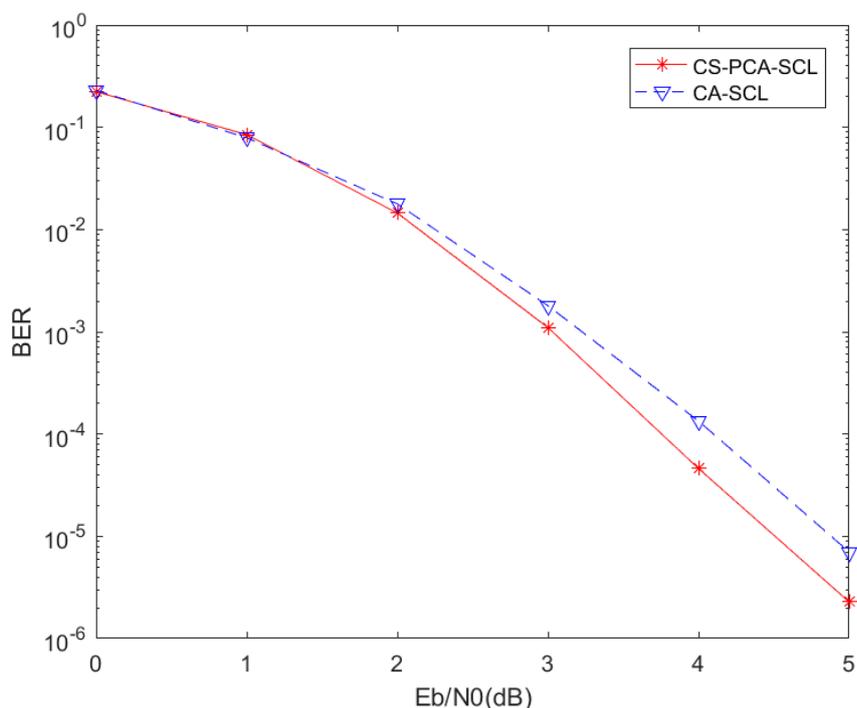


图 3-2 (128,68) 极化码在 CS-PCA-SCL 算法和 CA-SCL 算法下的译码性能比较
Figure 3-2 Comparison of decoding performance for (128,68) concatenated code under CS-PCA-SCL algorithm and CA-SCL algorithm

图 3-3 表示 (256,132) 的 CRC 与极化码的级联码在 CA-SCL 译码算法与 CS-PCA-SCL 译码算法在高斯信道下的仿真结果,其中两种译码算法设置的 CRC 校验位长度均为 4, SCL 算法中的搜索宽度 L 均设置为 4:

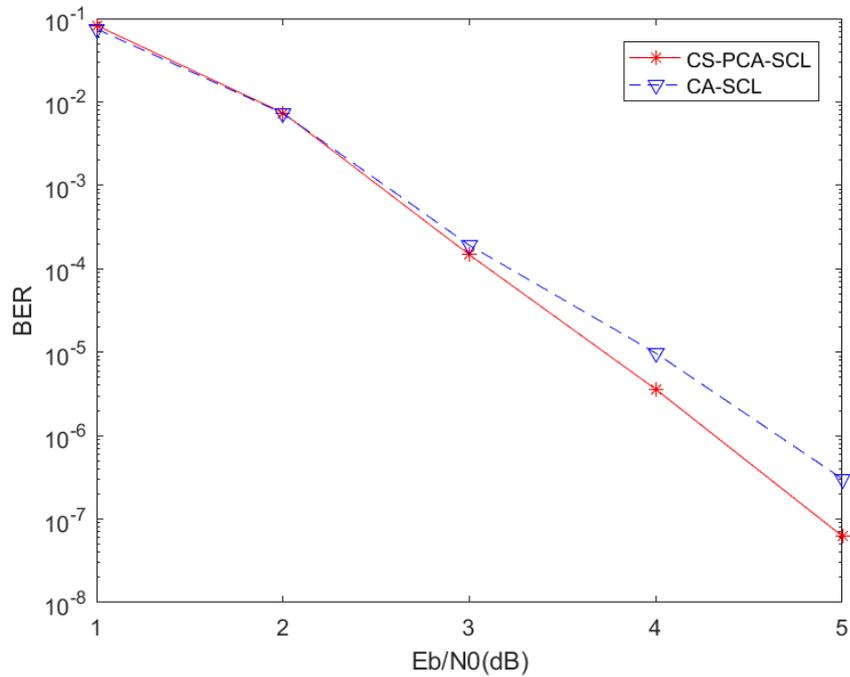


图 3-3 (256,132) 级联码在 CS-PCA-SCL 算法和 CA-SCL 算法下的译码性能比较
Figure 3-3 Comparison of decoding performance for (256,132) concatenated code under CS-PCA-SCL algorithm and CA-SCL algorithm

图 3-4 表示 (512,260) 的 CRC 与极化码的级联码在 CA-SCL 译码算法与 CS-PCA-SCL 译码算法在高斯信道下的仿真结果，其中两种译码算法设置的 CRC 校验位长度均为 4，SCL 算法中的搜索宽度 L 均设置为 4:

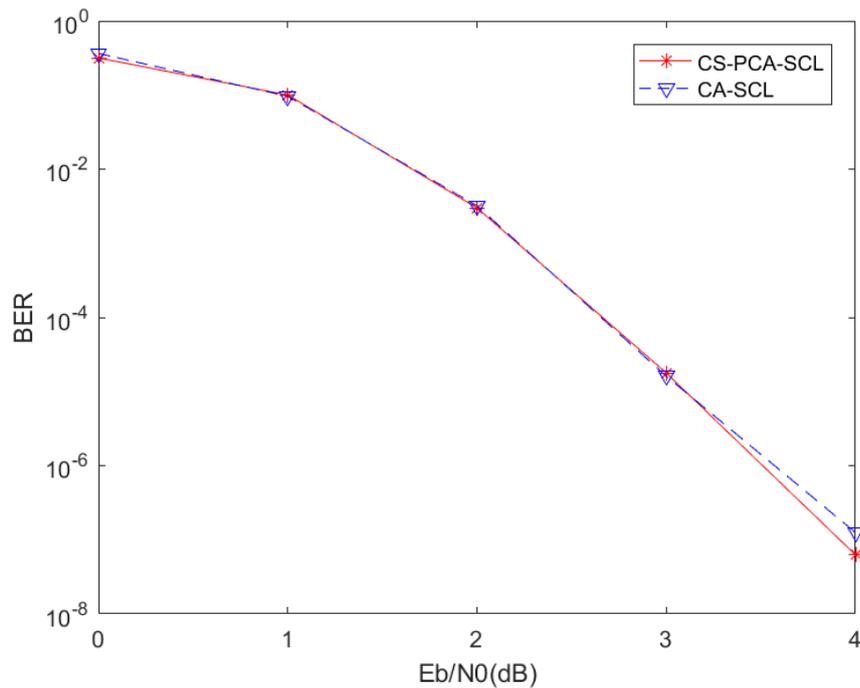


图 3-4 (512,260) 级联码在 CS-PCA-SCL 算法和 CA-SCL 算法下的译码性能比较
Figure 3-4 Comparison of decoding performance for (512, 260) concatenated codes under CS-PCA-SCL algorithm and CA-SCL algorithm

仿真结果的分析：从图 3-2，图 3-3，图 3-4 看出，就译码性能而言，CS-PCA-SCL 算法的译码性能要优于 CA-SCL 算法：在低信噪比下，两种译码算法的 BER 基本相等；在高信噪比下，CS-PCA-SCL 算法的译码性能明显要优于 CA-SCL 算法。比如 (128,68) 级联码在误比特率为 10^{-5} 时，CS-PCA-SCL 算法能够获得约 0.5 dB 的编码增益。而且在其他码长条件下，也都有不同程度的译码性能提升。究其原因，在于关键集合能够比较准确地估计译码发生错误的位置，只针对关键集合中的信息比特进行 CRC 校验就能够保证通过 CRC 校验即为译码正确。这样一来减少了 CRC 监督的信息个数，从而提高了 CRC 校验的准确性，使得在最后 L 条候选译码路径中进行选择时，能更大概率地选中正确译码路径。并且从仿真结果可以看出，在短码长下 CS-PCA-SCL 算法性能提升更明显，这是因为短码的误码类型更多，部分 CRC 校验和全局 CRC 校验的差异性才得以体现。

而从译码复杂度上看，由于只需要在关键集合的位置进行路径分裂，CS-PCA-SCL 译码算法大大降低了计算复杂度和空间复杂度。在此次实验中，不同码长下信息位个数和关键集合中元素的个数对比如表 3-2 所示，其中 N 表示码长， K 表示信息位个数， N_{cs} 表示关键集合元素个数：

表 3-2 不同码长下的信息位个数和关键集合元素个数
Table 3-2 Number of information bits and number of critical set elements under different code lengths

N	128	256	512
K	64	128	256
N_{cs}	20	38	65
N_{cs}/K	31.25%	29.9%	25.4%

由此可见，CS-PCA-SCL 算法在不同的码长下都减少了大约 2/3 的复杂度。

3.4 本章小结

本章介绍了译码可靠性测度——CRC。CRC 校验的优势在于校验原理较为简单，校验结果较为准确，便于硬件实现。CRC 校验算法在极化码译码中的应用较为广泛，文献[47]和文献[53]都提出了基于 CRC 校验的译码算法：CA-SCL 译码算法和单比特翻转译码算法。CRC 在极化码译码算法中主要是用于衡量译码路径的正确性。

本章在 CA-SCL 的基础上，提出 CS-PCA-SCL 译码算法，利用译错的信息比特大概率存在于关键集合中的特性，在 CA-SCL 译码算法基础上做出两点改进：一是只针对关键集合中的信息比特进行路径的分裂，可以大大减少译码复杂度；二

是使 CRC 只针对于关键集中的信息比特进行错误监督，减少了 CRC 监督的信息位个数，进一步提高了 CRC 校验的准确性，提升了译码性能。

4 译码可靠性测度：距离参数

在上一章中介绍了 CRC 校验算法，CRC 校验是一种比较常见的检测错误的方式，可以利用“正确码字可通过 CRC 校验”的先验信息对译码结果的可靠性进行度量，并基于 CA-SCL 译码算法研究出了 CS-PCA-SCL 译码算法，通过提高 CRC 校验的准确度，进一步改善了译码算法的性能。虽然 CRC 校验准确性较高，但仍存在一些问题。比如要使用 CRC 进行校验，就必须使用 CRC 级联码，会有编译码的开销，而且 CRC 校验位会占用较可靠子信道进行传输，使得信息比特占用信道的可靠性降低，从而损害译码性能。那么能否找到一种译码可靠性测度是不需要增加额外的冗余比特的呢？答案是可以的，本章研究了一种基于距离的译码可靠性测度，并将其命名为距离参数 (D)，将距离参数应用在单比特翻转译码算法中，替代 CRC 校验，进一步提高单比特翻转译码算法的性能。

4.1 距离参数

距离参数 (D) 可以衡量译码码字与信道接收向量之间的距离，选择距离参数值最小的译码路径符合最大似然译码规则。

令 $\mathbf{y} = (y_1, y_2, \dots, y_N)$ 为信道接收序列，则 $y_i = (-1)^{u_i} + w_i$, $i = 1, 2, \dots, N$ ，其中 w_i 是高斯随机变量：均值为零，方差为 σ^2 。最大似然是指：

$$P(Y|V_i) = \prod_{j=1}^N P(y_j/v_{ij}) \quad (4-1)$$

其中 $P(Y|V_i)$ 为似然函数， V 表示码字。若能在 2^K 个码字中选择某一个 V_i 能够使公式 (4-1) 取得最大值，则这种译码规则称为最大似然译码。对于 DMC 信道，最大似然译码是使译码错误概率最小的一种最佳译码准则。为了简化译码运算复杂度，最大似然译码可以等效为最小平方欧氏距离作为译码度量，也就是说接收序列 \mathbf{y} 被译为使平方欧氏距离最小的码字 \mathbf{v} 。而平方欧氏距离最小又可以等效为使下式最大：

$$m(\mathbf{y}, \mathbf{v}) = \sum_{i=0}^{N-1} v_i y_i \quad (4-2)$$

将此求和项称为 \mathbf{y} 与码字 \mathbf{v} 之间的相关测度。所以最大似然译码是将 \mathbf{y} 译为使 $m(\mathbf{y}, \mathbf{v})$ 最大的码字 \mathbf{v} 。

$m(\mathbf{y}, \mathbf{v})$ 求最大值等价于 $\sum_{i=0}^{N-1} (-1)^{v_i} \times y_i$ 这个求和项求最大值，又因为：

$$\begin{aligned}
\sum_{i=0}^{N-1} (-1)^{v_i} \times y_i &= \sum_{i=0}^{N-1} (-1)^{v_i} \times (-1)^{z_i} \times |y_i| \\
&= \sum_{i=0}^{N-1} (-1)^{v_i \oplus z_i} \times |y_i| \\
&= \sum_{v_i=z_i} |y_i| - \sum_{v_i \neq z_i} |y_i| \\
&= \sum_{i=0}^{N-1} |y_i| - 2 \sum_{v_i \neq z_i} |y_i|
\end{aligned}$$

其中 $z = (z_0, z_1 \dots z_{N-1})$ 表示硬判决接收字, $z_i = \text{sgn}(y_i)$ 。 $\sum_{i=0}^{N-1} |y_i|$ 为常量, 所以 $m(y, v)$ 最大等价于 $\sum_{v_i \neq z_i} |y_i|$ 求最小值。也就是说, 最大似然译码是将 y 译为使 $\sum_{v_i \neq z_i} |y_i|$ 最小的码字 v 。

给出距离参数的定义式如公式 (4-3) 所示:

$$D(v_i, z) \triangleq \sum_{j: v_i, j \neq z_j} |LLR_j| \quad (4-3)$$

其中 LLR 表示信道接收信息的似然比, 在高斯白噪声信道下, LLR 与信道接收信号 y 之间的关系如公式 (4-4):

$$LLR = -2y/\sigma^2 \quad (4-4)$$

距离参数也可以理解为反映译码码字与信道接收向量之间的距离关系的参数。

距离参数的值越小, 表示译码码字与信道序列硬判决码字之间的相关差异值越小。由以上分析可知, 挑选距离参数值最小的译码路径符合最大似然译码规则, 能够挑选出最优译码路径。

4.2 D-SCF 译码算法

选择距离参数值最小的码字符符合最大似然译码规则, 因此可以将距离参数应用在极化码译码算法中, 以对译码可靠性进行度量, 选择最优译码路径。本小节借助 Chase 算法的思想, 提出了基于距离参数的单比特翻转 (*D* aided SC flipping, *D*-SCF) 译码算法, 用距离参数替代 CRC 校验进行译码结果的选择。

4.2.1 Chase 算法的介绍

Chase 算法是个次最优译码过程, 是对广义最小距离 (generalized minimum

distance, GMD) 译码算法的推广^[63]。

Chase 译码算法的主要思想是通过找到最有可能的错误图样来纠正码字。错误图样是由接收信号中根据接收符号的可靠性选出来的最不可靠位生成的列表或集合，通常可靠性由接收信息似然比的幅度 $|LLR_i|$ 来判定，幅度值越小则越不可靠。每个错误图样被添加到硬判决接收字中，然后使用硬判决译码器进行译码，并将译码结果归入候选码字集中。最后选取与接收信号的距离值最小的候选码字作为译码结果。

Chase 译码算法是基于线性分组码的纠错特性而提出的，符合最大似然译码准则。具体工作原理如下：

假设来自信道的接收序列为 $y = (y_1, y_2, \dots, y_N)$ ， $y_i = (-1)^{u_i} + w_i$ ，其中 w_i 是均值为零，方差为 σ^2 的高斯随机变量， $i = 1, 2, \dots, N$ 。接收值的符号位代表硬判决接收字，如公式 (4-5)：

$$Y = (Y_0, Y_1, \dots, Y_N) \tag{4-5}$$

其中， $Y_i = \text{sgn}(y_i), 0 \leq i \leq N$

如图 4-1 所示，以 $C_1、C_2、C_3、C_4$ 为圆心的圆圈代表译码码字，以 Y 为圆心的圆圈代表接收序列的硬判决码字。译码码字圆圈都是以纠错能力 $t = [(d - 1)/2]$ 为半径的。

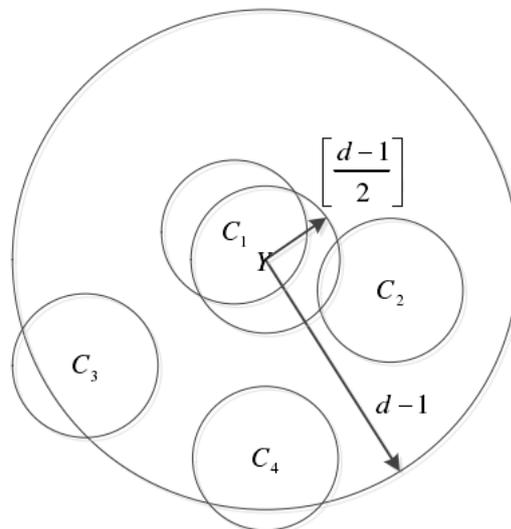


图 4-1 Chase 译码原理^[64]
Figure 4-1 Chase decoding principle^[64]

从图 4-1 中可以看出，各个码字所在的码字圆圈互不相交。Chase 译码算法就是在以码字 Y 为圆心，最大检错能力 $e = d - 1$ 为半径的圆圈内(d 为最小汉明距离)，

寻找与码字 Y 最近的码字作为译码输出。所以在图 4-1 中, C_1 就是通过 Chase 算法译码得到的最佳码字。

根据错误图样产生过程, Chase 将他的算法分为三类:

第 I 类: 检验距离接收字半径 $(d - 1)$ 以内的所有错误图样;

第 II 类: 检验位于 $[d/2]$ 个可靠性最低的位置的最多 $[d/2]$ 个错误的错误图样。与第 I 类相比, 第 II 类 Chase 算法需要测试的错误图样数大大减少, 而性能仅有轻微下降。

第 III 类: 检测在 i 个最不可靠位置为 1 的错误图样, i 为奇数且 $1 \leq i \leq d - 1$ 。

在上述三种算法中, 第 II 类 Chase 算法兼顾了性能与复杂度: 复杂度低, 性能好; 所以应用最为广泛。

第 II 类 Chase 算法的译码流程总结如下:

(1) 对于 $i = 1, 2, \dots, 2^t - 1$, 将错误图样 e_i 加到接收序列的硬判决码字上: $Y_i = e_i \oplus Y_0$;

(2) 在 t 个最不可靠位置产生错误图样, 这些位置的可靠性是最小的;

(3) 每个 Y_i 被输入到硬判决译码器中, 得到一个码字 $v_i, i = 1, 2, \dots, 2^t - 1$;

(4) 计算每个码字 v_i 和 Y 之间的欧氏距离, 选择欧氏距离最小的码字作为译码结果输出。

从算法的基本描述中可以明显看出, Chase 译码算法是一种软输入硬输出 (soft input hard output, SIHO) 译码算法, 且 Chase 算法具有接近最大似然译码性能的特性。

4.2.2 D -SCF 译码算法描述

信道经过信道联合和信道分裂后, 生成的虚拟子信道相互依赖: 序号大的子信道依赖于所有比它序号小的子信道。SC 译码算法正是充分利用了极化信道的这种依赖关系, 在进行译码时, 对当前比特进行判决需要假设之前比特全部判决正确, 所以前期一个估计错误的信息比特将可能在接下来译码中引起错误传递, 导致多个信息比特译码出错。因此尽可能早地发现并纠正译错的信息比特对于译码性能的提升是有益的。单比特翻转译码算法正是基于这样的思想。

在 2.2 节中已经介绍过单比特翻转译码算法, 从 2.2 节中可以注意到关键的两点:

(1) 在文献[53]中翻转位置的选择是基于虚拟信道的似然比值 LLR, 认为似然比幅度越小的信道译码结果越不可靠。设置翻转次数 T 后, 选择用于传输信息比特的虚拟子信道中似然比最小的 T 个信道作为翻转比特备选, 依次翻转。

(2) 在单比特翻转译码算法中，是通过 CRC 校验来验证 SC 译码结果是否正确以及判断是否应翻转信息比特继续译码。因此在编码的时候必须先将 CRC 校验位附在信息序列之后，生成 CRC 与极化码的级联码；然后将级联码送入信道中传输；最终依据 CRC 校验原理判断此次译码是否成功，若不成功，则继续翻转比特，重复 SC 译码过程。

针对单比特翻转译码算法的这两个特点，本小节提出来了基于距离参数的单比特翻转译码 (*D-SCF*) 算法，主要改进在于：

1) 使用关键集合缩小翻转比特的选择范围。

在 3.3 小节中，有详细介绍关键集合的原理及关键集合的寻找算法。从 3.3 节中我们知道，即使在低信噪比下，关键集合预测第一个出错的信息比特的准确率可以达到 99% 以上，而在较高信噪比下，预测准确度能达到 100%，因此可以将翻转比特的范围缩小至关键集合中的元素，而关键集合外的元素就算信道似然比再低，也不考虑将其翻转。在改进译码算法中，结合关键集合和单比特翻转算法，确定翻转次数 T 后，先找出关键集合，缩小翻转比特集合的范围，再从关键集合中找出似然比最小的 T 个位置，作为最终翻转比特备选集合，依次翻转。

2) 用距离参数替代 CRC 校验对译码码字进行选择。

在单比特翻转译码算法中，生成的 CRC 和极化码的级联码是在信息序列后添加 CRC 校验位实现的。由于极化码的信道极化特性，一般而言序号越靠后的虚拟信道信道容量越大，即信道可靠性越高。所以在传统的单比特翻转译码算法中，CRC 校验位占据了可靠的虚拟子信道传输，导致传输信息比特的信道的可靠性降低。因此，在改进算法中，使用距离参数替代 CRC 校验，选择距离参数值最小的译码路径作为最终译码结果输出。这样一来，可以不需要占用可靠信道用来传输额外的冗余位，不会影响译码性能；此外，选择距离参数值最小的译码路径作为最终译码结果输出，实际上就是选择与接收序列的硬判决码字欧氏距离最近的最佳码字，符合最大似然译码的思想。

总结基于距离参数的单比特翻转 (*D-SCF*) 译码算法流程如下：

第一步：对于码长为 N ，信息位个数为 K 的极化码进行虚拟子信道的可靠性衡量，并进行比特混合，选出冻结比特与信息比特的位置。依据关键集合寻找算法，找出对应的关键集合 CS 。此外对极化码编码之后的码字 X 进行 BPSK 调制之后送入信道，接收端得到受到噪声干扰后的信息序列 $y = (y_1, \dots, y_i, \dots, y_N)$, y_i 如公式 (4-6) 所示

$$y_i = 1 - 2X_i + w \quad (4-6)$$

公式中的 w 表示受到的噪声干扰， w 是均值为 0，方差为 σ^2 的高斯加性白噪声。

则对应的接收信号的对数似然比 LLR 为：

$$LLR = -\frac{2y}{\sigma^2}$$

根据公式 (4-7) 对 LLR 进行硬判决，得到信息序列的硬判决码字 Y ：

$$Y = \begin{cases} 1, & \text{if } LLR < 0 \\ 0, & \text{if } LLR > 0 \end{cases} \quad (4-7)$$

第二步：根据接收信号的对数似然比进行 SC 译码，得到第一次译码结果 U_1 。将 U_1 进行重编码，得到重编码后的码字 X_1 ，即：

$$X_1 = U_1 G \quad (4-8)$$

其中， G 为极化码的生成矩阵。并比较 X_1 与 Y ，计算 SC 译码结果的距离参数 D_1 ：

$$D_1 = \sum_{Y_i \neq X_{1i}} |LLR_i|, \quad i = 1, 2, \dots, K \quad (4-9)$$

第三步：给定翻转次数 T ，从关键集合中找出虚拟子信道中似然比最小的 T 个信息比特作为待翻转比特集合。每翻转一个比特并完成译码后，得到新的译码结果 U_j ，重复第二步中的重编码及距离参数的计算，最终得到 $T + 1$ 条译码路径及其对应的距离参数值。

第四步：比较所有候选路径的距离参数值，选择具有最小距离参数的译码路径作为最终的译码结果输出。

D -SCF 算法的中心思想与 Chase 算法类似，可以理解为都是通过找到最有可能的错误图样以纠正译码结果。但由于极化码结构的特殊性，在译码结束后直接得到信息比特序列，而不是码字，所以需要进行重编码，获得译码码字才能计算距离参数以决定最终译码输出结果。

用距离参数来衡量译码结果与用 CRC 校验算法的最大区别在于：距离参数是基于信道本身状态的一种衡量方式，不需要在信息序列之后增加额外的冗余位以构成 CRC 级联码用于传输。这样一来，不会因为 CRC 校验位占据虚拟子信道而导致传输信息比特的子信道的可靠性降低。所以，在某些情况下，使用距离参数来衡量译码结果是更加具有优势的。

4.3.3 仿真结果及分析

图 4-2 所示为极化码码长 $N = 128$ ，信息位数目 $K = 64$ 时，所提出的 D -SCF 译码算法与单比特翻转译码算法的性能对比。 D -SCF 译码算法与单比特翻转译码

算法的最大翻转次数均设置为 4，且翻转比特候选集合都是基于关键集合挑选的。其中单比特翻转译码算法中用到的 CRC 校验位长度为 4：

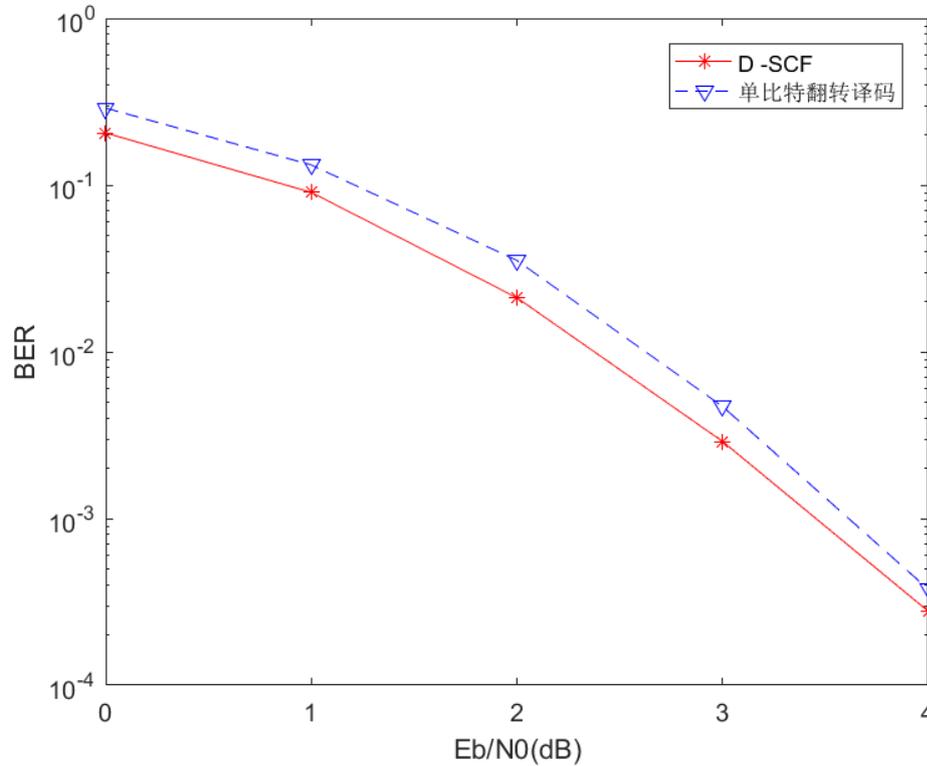


图 4-2 (128,64) 极化码 D -SCF 译码算法与单比特翻转译码算法仿真结果
Figure 4-2 Simulation results of (128,64) polar code under D -SCF decoding algorithm and single-bit flip decoding algorithm

图 4-3 所示为极化码码长 $N = 256$ ，信息位数目 $K = 124$ 时，所提出的 D -SCF 译码算法与单比特翻转译码算法的性能对比。 D -SCF 译码算法与单比特翻转译码算法的最大翻转次数均设置为 4，且翻转比特候选集合都是基于关键集合挑选的。其中单比特翻转译码算法中用到的 CRC 校验位长度为 4：

图 4-4 所示为极化码码长 $N = 512$ ，信息位数目 $K = 256$ 时，所提出的 D -SCF 译码算法与单比特翻转译码算法的性能对比。两者的最大翻转次数均设置为 4，且翻转比特候选集合都是基于关键集合挑选的。其中单比特翻转译码算法中用到的 CRC 校验位长度为 4：

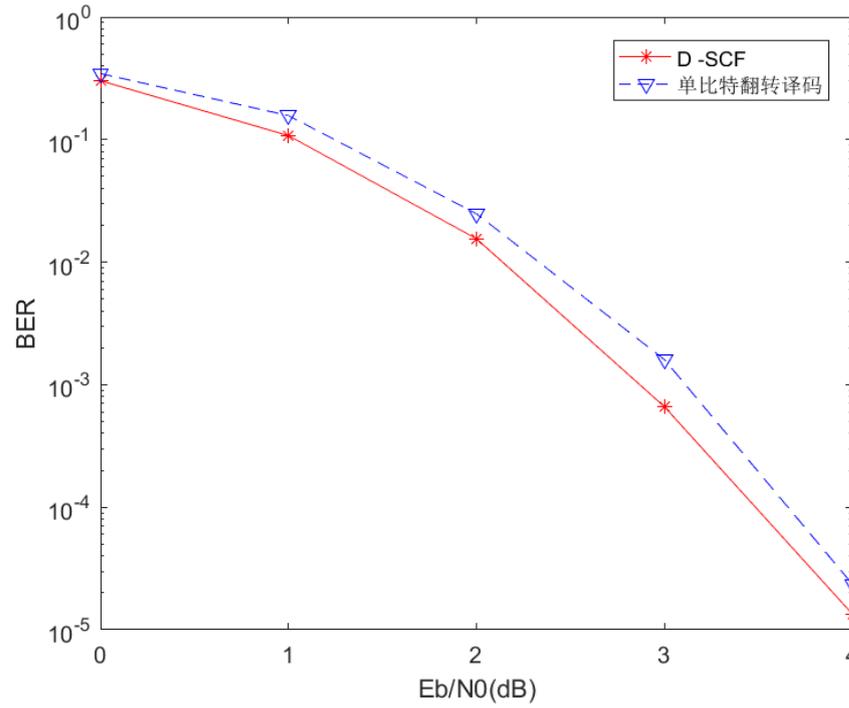


图 4-3 (256,124) 极化码 D -SCF 译码算法与单比特翻转译码算法仿真结果
Figure 4-3 Simulation results of (256,124) polar code under D -SCF decoding algorithm and single-bit flip decoding algorithm

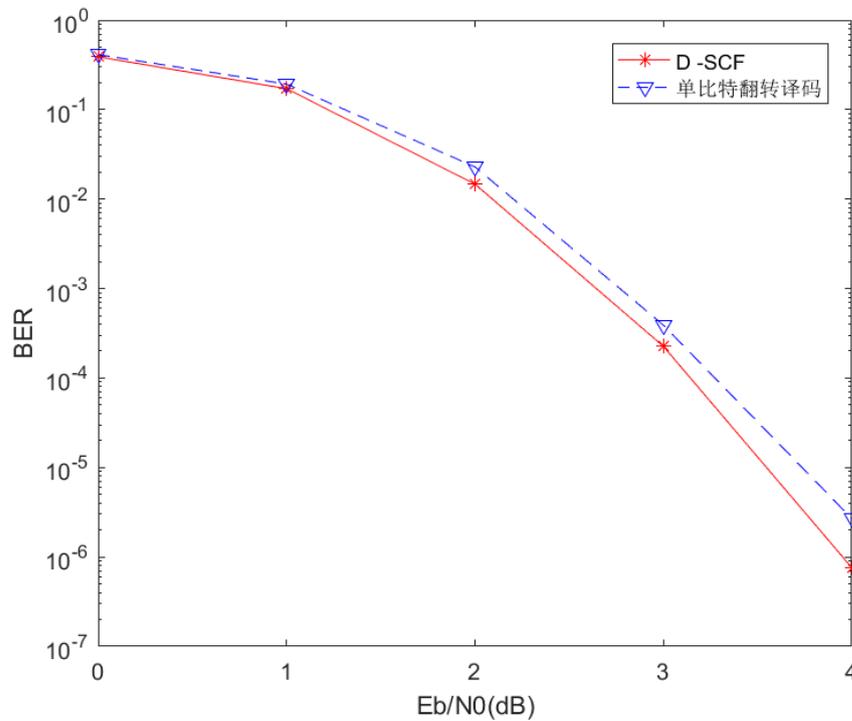


图 4-4 (512,256) 极化码 D -SCF 译码算法与单比特翻转译码算法仿真结果
Figure 4-4 Simulation results of (512,256) polar code under D -SCF decoding algorithm and single-bit flip decoding algorithm

从仿真结果分析， D -SCF 的性能要优于基于 CRC 校验的单比特翻转译码算法，且随着码长增长，提升性能越明显。性能提升的原因主要有以下三点：

①基于 CRC 校验的单比特翻转译码算法占据了最为可靠的几个子信道用于传输 CRC 校验位，使得传输信息比特的子信道的可靠性降低；而在 D -SCF 译码算法中，不会增加多余的冗余位来占据可靠信道进行传输；

②一旦翻转比特 u_i 后通过了 CRC 校验，则终止译码。有可能出现这样的情况：此次翻转 u_i 后并未译码正确，只是恰巧通过了 CRC 校验。但由于算法的设定，在此即终止译码，错失了正确翻转的译码路径。而对于 D -SCF 算法而言，会始终完成 $T + 1$ 次译码，并根据距离参数从中选择最优的译码路径，极大概率上选择正确译码结果。

③在基于 CRC 校验的单比特翻转译码算法中，若在 T 次翻转过后仍未有通过 CRC 校验的译码路径，则宣告译码失败，选择最初的 SC 译码结果输出。则存在下述情况：在 $T + 1$ 条候选路径中，虽无完全正确的译码路径，但存在某一条译码路径的错误比特数最少， D -SCF 将选择该路径作为最终译码结果输出，自然译码性能要优于基于 CRC 校验的单比特翻转译码算法。

所以，基于以上三种情况的分析， D -SCF 的译码性能会优于基于 CRC 校验的单比特翻转译码算法。

4.3 本章小结

本章研究了基于距离的译码可靠性测度——距离参数 D ，距离参数是基于信道本身而得到的，是通过衡量译码码字与接收序列硬判决码字之间的距离来选择译码路径，这种选择策略符合最大似然译码思想。

此外，本章还将距离参数应用在极化码译码算法中，提出了基于距离参数的单比特翻转译码算法(D -SCF)，此算法与传统单比特翻转译码算法最大的区别在于：利用距离参数替代 CRC 校验来进行译码路径的选择。仿真结果表明，基于距离参数的单比特翻转译码算法较基于 CRC 校验的单比特翻转译码算法性能更优，且从已做实验的结果观察，码长越长，基于距离参数的单比特翻转译码算法的性能提升更明显。

5 译码可靠性测度：冻结比特差异度

极化码中的冻结比特是指收发双方都已知的比特，一般设定为 0，可以辅助译码。本章即利用“冻结比特是收发双方都已知”的先验信息，提出另一种译码可靠性测度：冻结比特差异度 (δ)。与 CRC 和距离参数不同，获得 δ 的值不需要再经过复杂的计算，在译码过程中就能得到准确的 δ 值：通过统计译码时对冻结比特的硬判决结果与冻结比特设定值不一致的个数而得到。通过 δ 可以衡量译码结果的可靠性，并由此提出基于冻结比特差异度的自适应 SC-SCL 译码算法，提升译码性能。

5.1 定义

在极化码中，冻结比特是收发双方都已经约定好的比特，所以可以利用“冻结比特已知”的先验信息来判断译码结果的可靠性。本小节提出了一种新的译码可靠性测度，命名为冻结比特差异度 (δ)。该测度以对冻结比特的判定为基础，能够反映出此次译码结果的可靠程度，具体定义如下：

在冻结比特中根据相应的选择策略，挑选出关于冻结比特的特殊集合 Q ， Q 是指传输冻结比特的信道中可靠性相对较高的位置的集合。定义 Q 集合中虚拟子信道递归得到的 LLR 值的硬判决结果与规定的冻结比特设定值不一致的个数为冻结比特差异度 (δ)。

冻结比特差异度是利用冻结比特来对信道条件进行衡量，参数值越小，表明此次译码信道条件越好，译码结果越可靠。从定义中可以看出，并不需要对所有的冻结比特都进行比较，因为冻结比特所在的子信道原本就是较不可靠的虚拟信道，尤其当码长 N 较大时，部分虚拟信道的信道容量趋近于 0，即纯噪声信道。在这种信道传输信息，受到干扰导致译码出错的概率是非常大的，但并不一定会影响整体译码的正确性。所以根据所有的冻结比特译码硬判决结果与设定值不一致的个数来衡量译码结果的正确性是不明智的。在此，选取传输冻结比特的子信道中较为可靠的一部分信道构成特殊集合 Q ，统计选出来的这一部分子信道的 LLR 值的硬判决结果与冻结比特设定值不一致的个数，即为冻结比特差异度，以此作为判断此次译码结果是否可靠的依据。我们可以这样理解，这些信道可靠性较高，属于通过信道极化过程之后信道容量增大的虚拟信道，只是因为传输信息比特的信道数目有限，这些信道作为次优选择而“被迫”用于传输冻结比特。如果这些位置能保证译码结

果正确，那么更可靠的信道，即传输信息比特的信道出现译码错误的情况将大大降低。所以关键在于如何选择特殊集合 Q ，来提高冻结比特差异度判决的准确性。

5.2 特殊集合 Q 的选择策略

本小节将针对 Q 的选择，给出两种选择策略：巴氏参数选择法和 scaling exponent 选择法。

5.2.1 巴氏参数选择法

巴氏参数本身物理含义是指信道只传输 0 或只传输 1 时，进行最大似然判决错误概率的上限，所以可以利用巴氏参数对信道可靠性进行度量。综合信道极化的原理，第 i 个虚拟信道的巴氏参数 $Z(W_N^{(i)})$ 为：

$$Z(W_N^{(i)}) = \sum_{y_1^N \in Y^N} \sum_{u_1^{i-1} \in X^{i-1}} \sqrt{W_N^{(i)}(y_1^N, u_1^{i-1} | 0) W_N^{(i)}(y_1^N, u_1^{i-1} | 1)} \quad (5-1)$$

从直观感受上，巴氏参数和信道容量之间是反比的关系，即信道容量越大，其对应的巴氏参数值越小。事实上，Arikan 教授在论文中有明确给出两者之间的具体函数关系式如公式 (5-2) 和 (5-3)：

$$I(W) \geq \log \frac{2}{1+Z(W)} \quad (5-2)$$

$$I(W) \leq \sqrt{1 - Z(W)^2} \quad (5-3)$$

更小的巴氏参数值意味着信道更加可靠。当 $N \rightarrow \infty$ ， $Z(W_N^{(i)}) \rightarrow 0$ or $Z(W_N^{(i)}) \rightarrow 1$ 。 $Z(W_N^{(i)}) \rightarrow 0$ 意味着第 i 个虚拟信道是无噪声信道，信道可靠性高；相反， $Z(W_N^{(i)}) \rightarrow 1$ 意味着第 i 个虚拟信道是纯噪声信道，信道可靠性低。

在此规定巴氏参数小于 0.1 时，则认为该信道是比较可靠的，因此可以统计巴氏参数小于 0.1 时且用于传输冻结比特的虚拟子信道作为 Q 集合内的元素。

巴氏参数法选择 Q 集合的算法总结如下：

利用巴氏参数的迭代计算公式计算虚拟信道的巴氏参数值，统计巴氏参数小于 0.1 的虚拟子信道的数目 $N(Z)$ ，则 Q 集合中的元素的数目 $N(Q)$ 等于 $N(Z)$ 与信息比特数目 K 的差值，即：

$$N(Q) = N(Z) - K \quad (5-4)$$

在进行译码时，我们会按信道可靠程度进行从高到低的排序，选取前 K 个信道作为

传输信息比特的信道，信道可靠性排在 $K + 1 \sim K + N(Q)$ 中的信道即为 Q 集合中的元素。

5.2.2 计算巴氏参数的简便算法

利用巴氏参数法确定 Q 的范围，需要计算出各虚拟子信道的巴氏参数值。为了方便后续研究，在此提出巴氏参数的简便算法。

对于最简单的极化情况，即 $N = 2$ 时， $(W, W) \rightarrow (W_2^{(1)}, W_2^{(2)})$ ，虚拟信道 $W_2^{(1)}, W_2^{(2)}$ 的巴氏参数的计算规则如公式(5-5)，(5-6)所示：

$$Z(W_2^{(1)}) = Z(W)^2 \quad (5-5)$$

$$Z(W_2^{(2)}) \leq Z(W) \leq Z(W_2^{(1)}) \leq 2Z(W) - Z(W)^2 \quad (5-6)$$

特别地，当所在信道为 BEC 信道时，公式(5-6)可以直接取到等号，也就是说，在 BEC 信道下，可以通过递归运算来得到不同极化子信道的准确的巴氏参数值。对于其他信道的处理，可以假定取到等号，得到巴氏参数的近似值。我们在 BEC 的信道条件下对巴氏参数的简便算法进行阐述。

从公式(5-5)，(5-6)中可以得知，原始信道在经过一次信道极化之后，巴氏参数值发生变化，一个信道的巴氏参值变得更大，另一个信道的巴氏参数值变小。但是变化幅度 ϕ 是一致的，均为 $Z(W) - Z(W)^2$ 。即

$$\phi(Z(W)) = Z(W) - Z(W)^2 \quad (5-7)$$

很容易证明：

$$\phi(0.5 + Z(W)) = \phi(0.5 - Z(W)) \quad (5-8)$$

说明变换幅度关于巴氏参数的函数是一个轴对称函数，对称轴为 $Z(W) = 0.5$ 。所以给定任意两个关于 0.5 对称的巴氏参数值，则经过一次极化过程后巴氏参数的变换幅度是一致的，又因为巴氏参数的取值范围为 $[0,1]$ ，所以把巴氏参数的这个特性称之为互补性。巴氏参数的互补性是简化算法的理论基础。

极化码是一种多级编码方式，例如 $N = 8$ 时的编码过程如图 5-1 所示。

从图 5-1 中可以看出，极化码是一种多级编码，共有 $\log_2 N$ 级编码，每一级包含 $N/2$ 个极化单元。

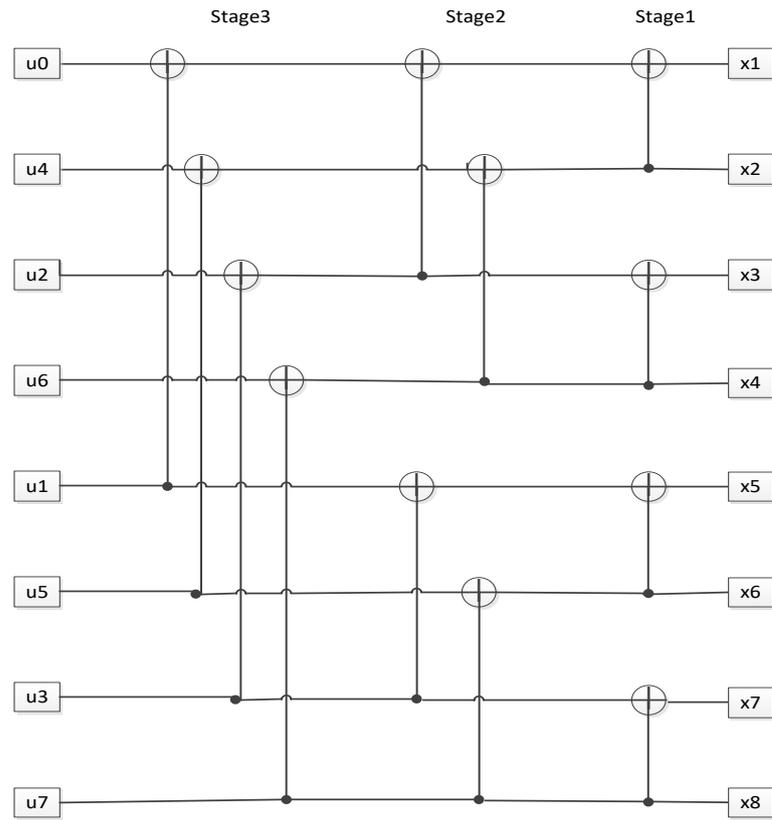


图 5-1 $N = 8$ 时极化码编码过程^[13]
Figure 5-1 Polar code encoding process when $N = 8$ ^[13]

在此给出如下定义：

过程 0：是指经过极化单元上支路的极化过程，巴氏参数值增大；

过程 1：是指经过极化单元下支路的极化过程，巴氏参数值减小；

至此，输入比特 $u_i(i_1 \dots i_{n-1} i_n)$ 可以用“0”和“1”来表示，其中 i_n 表示第 i 阶段的过程。参照图 5-1，则可以将输入比特可以表示为：

$$u_0(000); u_1(001); u_2(010); u_3(011);$$

$$u_4(100); u_5(101); u_6(110); u_7(111);$$

在接下来的分析中将使用过程 0 和过程 1 来表示译码比特 u_i 。

根据巴氏参数的互补性可以实现以下两种情况下的巴氏参数的简便运算：

(1) 计算互补信道条件下的虚拟子信道的巴氏参数

比如，在码长相等的条件下，如果我们已知 $Z(W) = 0.1$ ，即原始信道的巴氏参数的值为 0.1 时的虚拟子信道的巴氏参数值。基于巴氏参数的互补性，可以通过简单的加减运算得到 $Z(W) = 0.9$ 时的虚拟子信道的巴氏参数值。以 $N = 8$ 为例：已知当 $Z(W) = 0.1$ 时：

$$Z_1[u_{0(000)}] = 0.5695;$$

$$Z_1[u_{1(001)}] = 0.1183;$$

$$Z_1[u_{2(010)}] = 0.0709;$$

$$Z_1[u_{3(011)}] = 0.0013;$$

$$Z_1[u_{4(100)}] = 0.0309;$$

$$Z_1[u_{5(101)}] = 0.0004;$$

$$Z_1[u_{6(110)}] = 0.00019;$$

$$Z_1[u_{7(111)}] = 0.00000001$$

则根据互补性可以推得 $Z(W) = 0.9$ 时：

$$Z[u_{0(000)}] = 0.9 + (0.1 - Z_1[u_{0(111)}]) = 0.9999;$$

$$Z[u_{1(001)}] = 0.9 + (0.1 - Z_1[u_{0(110)}]) = 0.9998;$$

$$Z[u_{2(010)}] = 0.9 + (0.1 - Z_1[u_{0(101)}]) = 0.9996;$$

$$Z[u_{3(011)}] = 0.9 + (0.1 - Z_1[u_{0(100)}]) = 0.9691;$$

$$Z[u_{4(100)}] = 0.9 + (0.1 - Z_1[u_{0(011)}]) = 0.9987;$$

$$Z[u_{5(101)}] = 0.9 + (0.1 - Z_1[u_{0(010)}]) = 0.9291;$$

$$Z[u_{6(110)}] = 0.9 + (0.1 - Z_1[u_{0(001)}]) = 0.8817;$$

$$Z[u_{7(111)}] = 0.9 + (0.1 - Z_1[u_{0(000)}]) = 0.4341.$$

即通过简单的加减运算便可由 $Z(W) = 0.1$ 时子信道的巴氏参数值得到 $Z(W) = 0.9$ 时子信道的巴氏参数值。这并不只是个例，只要两个原始信道的巴氏参数值互补，则如果已知一种情况下的虚拟信道的巴氏参数值，就可以通过简单的加减运算得到另一情况下虚拟信道的巴氏参数值。

(2) 计算某一特定信道条件下的虚拟子信道的巴氏参数

若码长 $N = 8$ 时，在编码过程中巴氏参数的变化如图 5-2 所示：

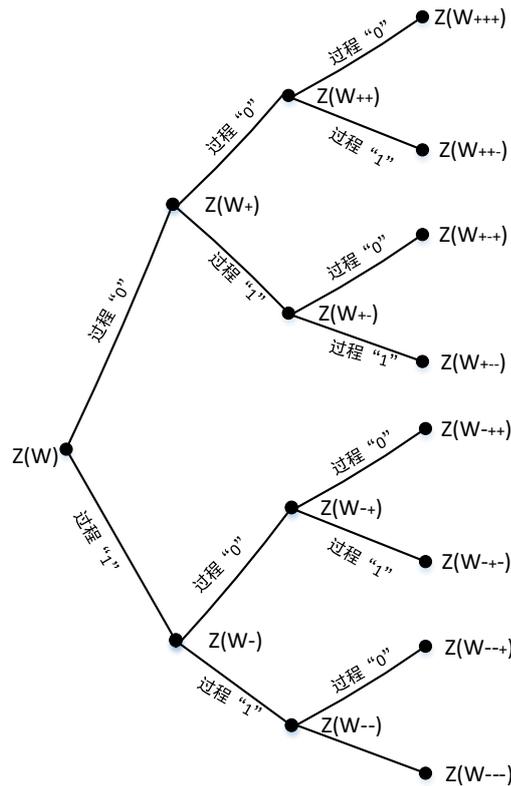


图 5-2 $N = 8$ 时，编码过程中巴氏参数的变化
Figure 5-2 When $N=8$, the change of Bhattacharyya parameters during the encoding process

图中“+”表示巴氏参数值的增加，“-”表示巴氏参数值的减少。从图中可以看出，虚拟子信道的巴氏参数值可以由提出的变换幅度 ϕ 计算得到。假设信道的巴氏参数值为 Z ，则：

$N=2$ 的情况下，

$$u_0: Z_+ = Z + \phi(Z)$$

$$u_1: Z_- = Z - \phi(Z)$$

$N=4$ 的情况下，

$$u_{0(00)}: Z_{++} = Z_+ + \phi(Z_+),$$

$$u_{1(01)}: Z_{+-} = Z_+ - \phi(Z_+),$$

$$u_{2(10)}: Z_{-+} = Z_- + \phi(Z_-),$$

$$u_{3(11)}: Z_{--} = Z_- - \phi(Z_-);$$

从这两组公式可以看出，利用变换幅度 ϕ 计算巴氏参数，能够减少乘法运算的次数从而降低运算复杂度。比如 $N = 8$ 时，要得到全部虚拟子信道的巴氏参数值，按照原始计算方法，需要进行一共 14 次乘法运算和 7 次加法运算；而利用变换幅

度 \emptyset ，则只需要计算 7 次乘法运算。特别地，当原始信道 $Z(W) = 0.5$ 时，只需要计算 $N/2$ 的虚拟子信道的巴氏参数值，剩下一半信道的巴氏参数值可由巴氏参数的互补性得到。比如， $N = 8$ ，且 $Z(W) = 0.5$ 时，可以计算得到：

$$Z[u_{0(000)}] = 0.9961;$$

$$Z[u_{1(001)}] = 0.8789;$$

$$Z[u_{2(010)}] = 0.8086;$$

$$Z[u_{3(011)}] = 0.3164;$$

则根据互补性：

$$Z[u_{4(100)}] = 0.5 + (0.5 - Z[u_{3(011)}]) = 0.6836$$

$$Z[u_{5(101)}] = 0.5 + (0.5 - Z[u_{2(010)}]) = 0.1914$$

$$Z[u_{6(110)}] = 0.5 + (0.5 - Z[u_{1(001)}]) = 0.1211$$

$$Z[u_{7(111)}] = 0.5 + (0.5 - Z[u_{0(000)}]) = 0.0039$$

可以明显看出，剩下的 $N/2$ 个子信道的巴氏参数值，我们可以只进行加减运算得到，因此能进一步减少一半的运算量。

将两种情况下，提出的巴氏参数简化算法和原始巴氏参数递归算法的运算复杂度对比如表5-1和表5-2所示，可以看出提出的算法简化了虚拟子信道的巴氏参数计算：

表 5-1 计算互补信道条件下的虚拟子信道的巴氏参数的复杂度

Table 5-1 The calculation complexity of Bhattacharyya parameters of virtual sub-channels under complementary channel conditions

	乘法	加法
原算法	$2(N - 1)$	$N - 1$
简化算法	0	N

表 5-2 计算某一特定信道条件下的虚拟子信道的巴氏参数的复杂度

Table 5-2 The calculation complexity of Bhattacharyya parameters of the virtual sub-channel under a certain channel condition

	乘法	加法
原算法	$2(N - 1)$	$N - 1$
简化算法	$N - 1$	$3(N - 1)$

5.2.3 巴氏参数选择法的效果

根据 5.2.1 节介绍的巴氏参数法选择 Q 集合的算法，进行了码长 $N = 256$ ，信息比特数 $K = 128$ 的极化码在高斯信道下的仿真，给出了冻结比特差异度值对于译码结果衡量的准确性的仿真结果。

当信噪比为 4 dB 时，巴氏参数值小于 0.1 的虚拟信道数 $N(Z)$ 为 161 个，所以 Q 集合中的信道数目 $N(Q) = 161 - 128 = 33$ 个。因此选取信道可靠性排序中排在第 129~161 位的信道作为 Q 集合中的位置，统计这些位置的 LLR 值的硬判决结果和冻结比特设定值不一致的个数即为冻结比特差异度 δ 。一共仿真了 16000 条数据，将 δ 值以及在译码中对应出现的次数 N_f 统计如表 5-3 所示，并给出在当前 δ 下出现译码错误的次数 N_e ，计算出在特定的 δ 下译码错误概率 P 。

表 5-3 信噪比为 4 dB 时 δ 及对应的译码情况

Table 5-3 δ and corresponding decoding conditions when signal-to-noise ratio is 4 dB

δ	0	1	2	3	4	5
N_f	15398	529	60	10	1	2
N_e	0	1	1	5	1	2
P	0	0.0018	0.0166	0.5	100%	100%

当信噪比为 3 dB 时，巴氏参数值小于 0.1 的虚拟信道数 $N(Z)$ 为 135 个，所以 Q 集合中的信道数目 $N(Q) = 135 - 128 = 7$ 个。因此选取信道可靠性排序中排在第 129~135 位的信道作为 Q 集合中的位置，统计这些位置的 LLR 值的硬判决结果和冻结比特设定值不一致的个数即为冻结比特差异度 δ 。一共仿真了 9318 条数据，将 δ 值以及在译码中对应出现的次数 N_f 统计如表 5-4 所示，并给出在当前 δ 下出现译码错误的次数 N_e ，计算出在特定的 δ 下译码错误概率 P 。

表 5-4 信噪比为 3 dB 时 δ 值及对应的译码情况

Table 5-4 δ and corresponding decoding conditions when signal-to-noise ratio is 3 dB

δ	0	1	2	3	5
N_f	9099	196	13	8	2
N_e	45	36	9	8	2
P	0.0049	0.18	0.69	100%	100%

从表 5-3，表 5-4 中的仿真结果可以看出：码长 $N = 256$ ，信息比特数 $K = 128$ 的极化码在高斯信道下进行仿真时，用巴氏参数法选择 Q 集合的前提下， $\delta \leq 1$ 表示译码结果大概率未出现错误，可靠性高； $\delta \geq 3$ 表示译码大概率出错，译码结果不可靠。

5.2.4 scaling exponent 选择法

本文提出的另一种 Q 集合的选择策略是基于 scaling exponent 的，scaling exponent 描述的是码率 R ，码长 N ，和误块率 P_e 三者之间的衡量关系。

定义 $P_N(R, C)$ 是指码长为 N ，码率为 R 的码被传输在信道容量为 C 的信道上用于通信时的块错误概率。假设存在函数 f 和参数 $\mu > 0$ ，对于误块率 P 存在如下关系^[65]：

$$\lim_{N \rightarrow \infty: N^{\frac{1}{\mu}}(C-R)=z} P_N(R, C) = f(z) \quad (5-9)$$

公式 (5-9) 即 scaling law。其中 f 称之为母函数，参数 μ 称之为 scaling exponent。注意，当码长趋于无限长时：

$$C - R \approx N^{-1/\mu} \quad (5-10)$$

Korada 和 Montanari 等人证明了极化码的误块率下界满足 scaling law，并且给出了极化码在不同信道下的 scaling exponent 值^[65]。具体地：对于 BEC 信道而言， $\mu \approx 3.6261$ ；对于二进制高斯白噪声信道（BAWGNC）而言， $\mu \approx 4.007$ 。

在文献^[65]进一步指出，对于随机码，要实现可靠通信，码率应比信道容量低 $N^{-1/2}$ ，即满足公式 (5-11)：

$$C - R \approx N^{-1/2} \quad (5-11)$$

而对于 AWGN 信道下的极化码而言，想要实现可靠通信，则信道容量，码率和码长三者之间的关系应满足公式 (5-12)

$$C - R \approx N^{-1/4.007} \quad (5-12)$$

所以利用公式 (5-12)，我们可以总结 scaling exponent 选择法的思想如下：

根据 scaling law，可以得出码长为 N ，信道容量为 C 的条件下的最佳码率 R' ，从而得到最合适的信息位个数 $K' = N \times R'$ ，则 Q 集合中的信道数目 $N(Q)$ 等于 K' 减去现在的信息位个数 K ，即：

$$N(Q) = K' - K \quad (5-13)$$

在进行译码时，我们会按信道可靠程度进行从高到低的排序，选取前 K 个信道作为传输信息比特的信道，信道可靠性排在 $K + 1 \sim K + N(Q)$ 中的信道即为 Q 集合中的信道。

5.2.5 scaling exponent 选择法的效果

根据上面介绍的基于 scaling exponent 选择 Q 集合的算法,进行了码长 $N = 256$,信息比特数 $K = 128$ 的极化码在高斯信道下的仿真,给出了冻结比特差异度值对于译码结果衡量的准确性的仿真结果。

当信噪比为4 dB时,查阅文献可知^[66],在4 dB下,对应的信道容量 $C \approx 0.95$ 。根据公式(5-12)可知,要保证可靠的通信,最适合的码率 $R' \approx 0.69$,所以根据 scaling 规则,在此条件下最合适的信息位个数 $K' = 256 \times R' \approx 179$ 。因此 Q 集合中的数目 $N(Q)=179-128=51$ 个,选取信道可靠性排序中排在第 129~179 位的信道作为 Q 范围中的位置,统计这些位置的 LLR 值的硬判决结果和冻结比特设定值不一致的个数即为冻结比特差异度 δ 。一共仿真了 16000 条数据,将 δ 值以及在译码中对应出现的次数 N_f 统计如表 5-5 所示,并给出在当前 δ 下出现译码错误的次数 N_e ,计算出在特定的 δ 下译码错误概率 P 。

表 5-5 信噪比为4 dB时 δ 值及对应的译码情况

Table 5-5 δ and corresponding decoding conditions when signal-to-noise ratio is 4 dB

δ	0	1	2	3	4	5	6	7	8	10
N_f	12759	2321	667	192	45	14	6	2	2	2
N_e	0	0	0	1	0	1	0	0	1	2
P	0	0	0	0.005	0	0.071	0	0	0.5	100%

当信噪比为3 dB时,查阅文献可知^[66],在3 dB下,对应的信道容量 $C \approx 0.9$ 。根据公式(5-12)可知,要保证可靠的通信,最适合的码率 $R' \approx 0.6494$,所以根据 scaling law,在此条件下最合适的信息位个数 $K' = 256 \times R' \approx 166$ 。所以 Q 集合中的数目 $N(Q)=166-128=38$ 个,因此选取信道可靠性排序中排在第 129~166 位的信道作为 Q 集合中的位置,统计这些位置的 LLR 值的硬判决结果和冻结比特设定值不一致的个数即为冻结比特差异度 δ 。一共仿真了 8623 条数据,将 δ 值以及在译码中对应出现的次数 N_f 统计如表 5-6 所示,并给出在当前 δ 下出现译码错误的次数 N_e ,计算出在特定的 δ 下译码错误概率 P 。

表 5-6 信噪比为3 dB时 δ 及对应的译码情况

Table 5-6 δ and corresponding decoding conditions when signal-to-noise ratio is 3 dB

δ	0	1	2	3	4	5	6	7	8	≥ 9
N_f	5419	1926	757	286	124	49	14	12	7	29
N_e	1	2	5	9	20	14	5	10	5	29
P	0.00018	0.001	0.007	0.005	0.16	0.29	0.36	0.83	0.71	100%

从表 5-5, 表 5-6 中的仿真结果可以看出: 码长 $N = 256$, 信息比特数 $K = 128$ 的极化码在高斯信道下进行仿真时, 用 `scaling exponent` 法选择 Q 集合的前提下, $\delta \leq 1$ 表示译码结果基本未出现错误, 可靠性高; $\delta \geq 7$ 表示译码大概率出错, 译码结果不可靠。

5.2.6 两种选择算法的对比

针对于 $N = 256$, 信息比特数 $K = 128$ 的极化码在高斯信道下进行仿真, 观察表 5-3, 表 5-4, 表 5-5, 表 5-6 中的数据综合对比两种 Q 集合选择算法。

基于巴氏参数的选择算法: 在 4 dB 的条件下, $\delta \leq 1$ 时, 可以基本认为此次译码结果可靠; 在 3 dB 的条件下, $\delta \leq 1$ 时, 可以大概率认为此次译码结果可靠, 但冻结比特差异度等于 1 的判断准确率要低一些。

基于 `scaling exponent` 的选择算法: 在 4 dB 的条件下, $\delta \leq 1$ 时, 可以基本认为此次译码结果可靠; 在 3 dB 的条件下, $\delta \leq 1$ 时, 也可以基本认为此次译码结果可靠。

在高信噪比的情况下, 两种算法的性能差异并不是很大, 但在低信噪比的条件下, 基于 `scaling exponent` 的选择算法的精度要高一些。究其原因, 巴氏参数在高斯信道下只能得到虚拟子信道的近似巴氏参数值, 在进行子信道可靠性排序时不够准确, 导致 Q 集合的选择出现误差。

由仿真结果的对比发现, 基于 `scaling exponent` 的选择算法性能更加稳定, 在接下来提出的基于冻结比特差异度的 SC-SCL 自适应译码算法中将采取 `scaling exponent` 选择法来选择 Q 集合。

5.3 SC-SCL 自适应算法

基于以上分析, 可以给出如下定义: 若冻结比特差异度 $\delta \leq 1$, 则认为此次译码结果正确; $\delta > 1$, 则认为译码结果不准确, 需要采取别的措施来提高译码结果的可靠性。在此定义的基础上, 本小节提出了一种基于冻结比特差异度的 SC-SCL 自适应译码算法。

5.3.1 算法描述

SC 译码算法比较大的问题是“一锤子买卖”, 即一旦译完最后一个信息比特就

结束译码，无法判断译码结果的可靠性。那么如何能够有效地衡量 SC 译码算法的可靠性呢？通过前面的分析可以看出，冻结比特差异度能够衡量译码结果是否正确，而且不同于第三章，第四章提出的 CRC 和距离参数，冻结比特差异度不需要再进行额外的计算，直接在译码过程中就能得到。本小节提出基于冻结比特差异度的 SC-SCL 译码算法，在 SC 译码结束之后通过冻结比特差异度判断此次译码结果的可靠性，若不可靠，则采用 SCL 译码算法继续译码。

将基于冻结比特差异度的 SC-SCL 译码算法流程总结如下：

- (1) 按照原始 SC 译码算法进行译码，得到最后的译码比特；
- (2) 依据选出的 Q 集合确定此次 SC 译码的 δ 的值，若 $\delta \leq 1$ ，则认为此次 SC 译码结果可靠，输出 SC 译码结果作为此次译码最终结果。否则，转入步骤 (3)
- (3) 此时 $\delta > 1$ ，不能确信此次 SC 译码结果正确。为了保险起见，将启用 SCL 译码算法，将搜索宽度 L 设定为 4，进行 SCL 译码，最终译码结果以 SCL 译码算法的结果为准。

针对冻结比特差异度中的 Q 集合的确定，在 5.2 节中给出了两种算法。从仿真结果来看，两种算法的性能差异并不是很大，但基于 scaling exponent 的选择算法更稳定一些，尤其在低信噪比的条件下，选择范围更加准确。所以采用 scaling exponent 选择算法确定 Q 集合。

5.3.2 仿真结果及分析

由于 SC 译码算法不会对译码结果进行校验，在译码出现错误时也就不了了之了，并不会采取额外的措施改进译码性能。所以本章用冻结比特差异度 δ 对 SC 译码结果可靠性进行衡量，提出了 SC-SCL 自适应译码算法。

如图 5-3 是当码长 $N = 128$ ，信息位个数 $K = 64$ 时，提出的基于冻结比特差异度的 SC-SCL 译码算法与 SC 译码算法及 SCL 译码算法的仿真结果对比。其中，SCL 译码算法的搜索宽度 $L = 4$ 。

将每个信噪比下总仿真码字个数 N_t 以及 SC-SCL 算法中只进行 SC 译码的次数 N_a 的具体数据总结如表 5-7 所示。并且，通过 N_a/N_t 可以很方便地得出在 SC-SCL 译码算法中只进行 SC 译码次数的占比 $A\%$ 。

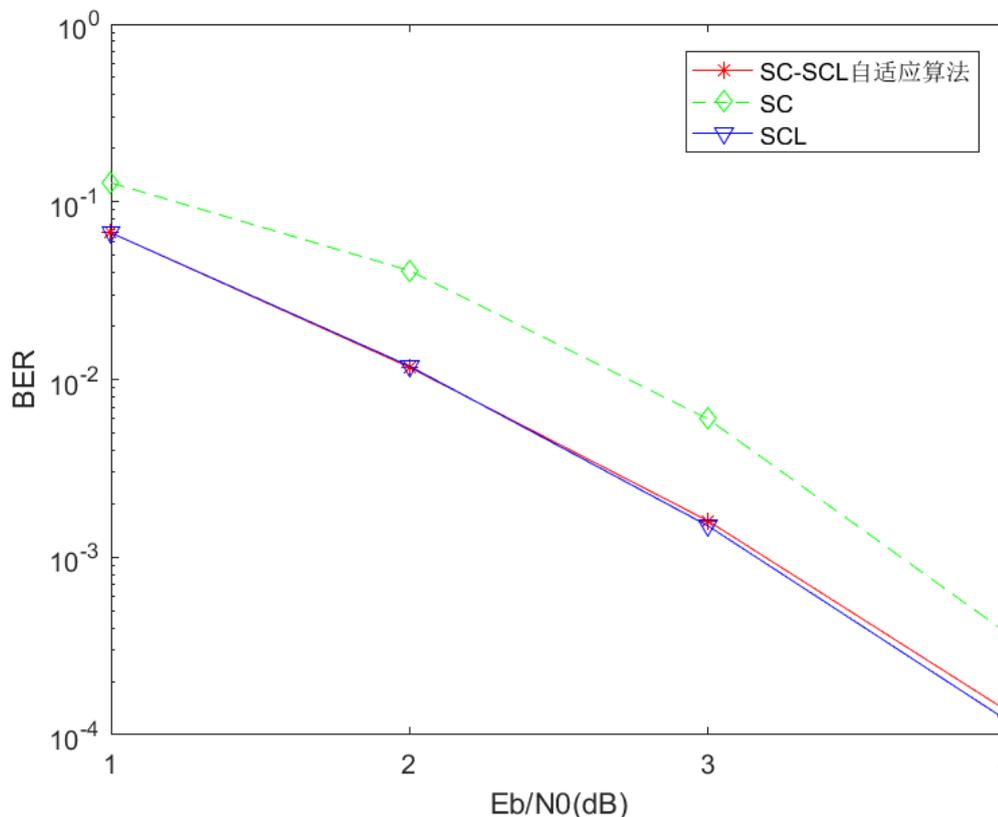


图 5-3 (128,64) 极化码 SC-SCL 译码算法仿真结果

Figure 5-3 Simulation results of (128,64) polar code under SC-SCL decoding algorithm

表 5-7 (128,64) 极化码 SC-SCL 算法中只执行 SC 算法的比例

Table 5-7 The proportion of SC-SCL algorithm that only executes SC algorithm for (128,64) polar code

信噪比 (dB)	1	2	3	4
N_t	500	1728	9806	102522
N_a	114	999	8659	101331
A%	22.8%	57.8%	88.3%	98.8%

如图 5-4 是当码长 $N = 256$ ，信息位个数 $K = 128$ 时，提出的基于冻结比特差异度的 SC-SCL 译码算法与 SC 译码算法及 SCL 译码算法的仿真结果，其中，SCL 译码算法的搜索宽度 $L = 4$ 。

将每个信噪比下总仿真码字个数 N_t 以及 SC-SCL 算法中只进行 SC 译码的次数 N_a 的具体数据总结如表 5-8 所示。并且，通过 N_a/N_t 可以很方便地得出在 SC-SCL 译码算法中只进行 SC 译码次数的占比 $A\%$ 。

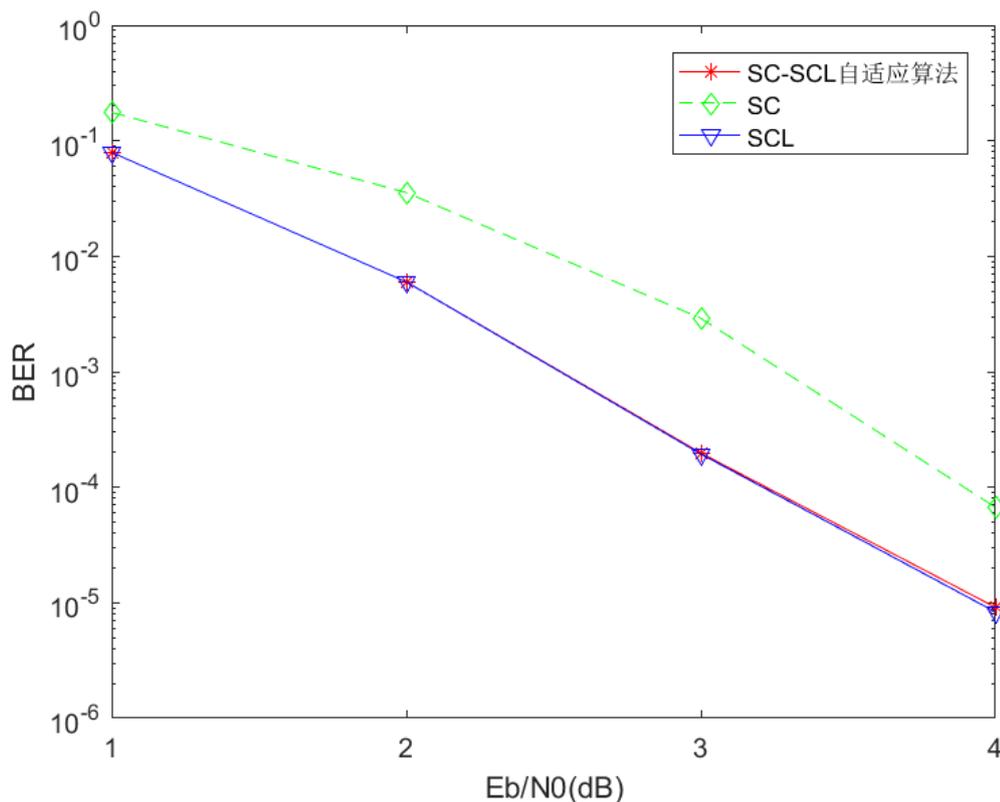


图 5-4 (256,128) 极化码 SC-SCL 译码算法仿真结果

Figure 5-4 Simulation results of (256,128) polar code under SC-SCL decoding algorithm

表 5-8 (256,128) 极化码 SC-SCL 算法中只执行 SC 算法的比例

Table 5-8 The proportion of SC-SCL algorithm that only executes SC algorithm for (256,128) polar code

信噪比 (dB)	1	2	3	4
N_t	500	3400	62605	724137
N_a	14	1207	52957	717721
$A\%$	2.8%	35.5%	84.6%	99.1%

如图 5-5 是当码长 $N = 512$ ，信息位个数 $K = 256$ 时，提出的基于冻结比特差异度的 SC-SCL 译码算法与 SC 译码算法及 SCL 译码算法的仿真结果，其中，SCL 译码算法的搜索宽度 $L = 4$ 。

将每个信噪比下总仿真码字个数 N_t 以及 SC-SCL 算法中只进行 SC 译码的次数 N_a 的具体数据总结如表 5-9 所示。并且，通过 N_a/N_t 可以很方便地得出在 SC-SCL 译码算法中只进行 SC 译码次数的占比 $A\%$ 。

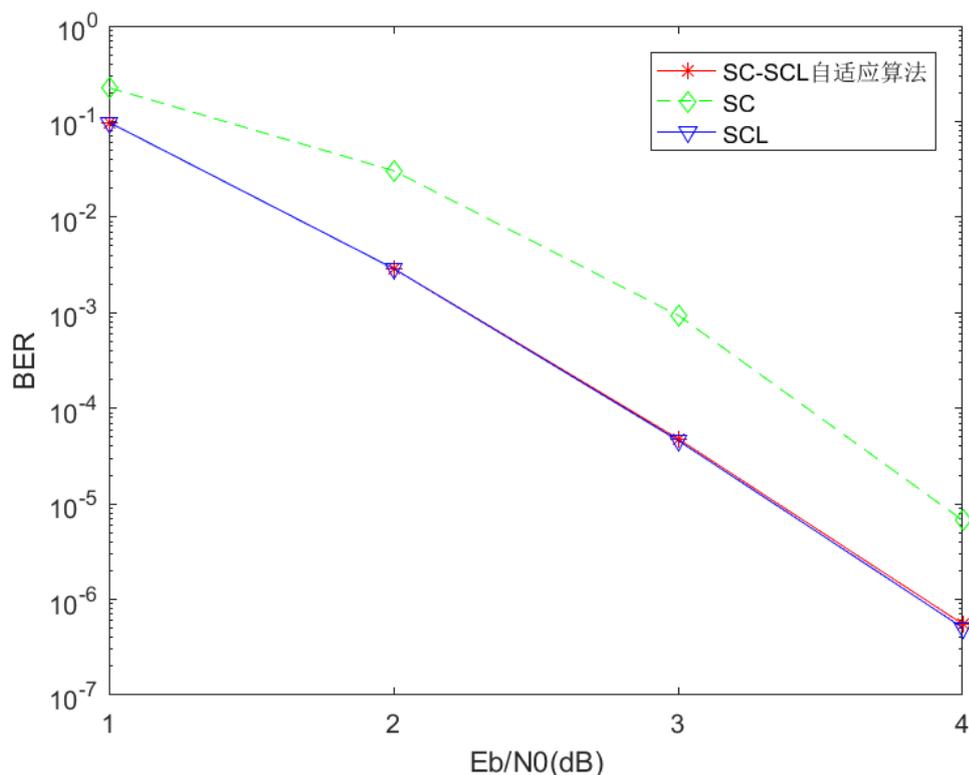


图 5-5 (512,256) 极化码 SC-SCL 译码算法仿真结果

Figure 5-5 Simulation results of (512,256) polar code under SC-SCL decoding algorithm

表 5-9 (512,256) 极化码 SC-SCL 算法中只执行 SC 算法的比例

Table 5-9 The proportion of SC-SCL algorithm that only executes SC algorithm for 512,256 polar code

信噪比 (dB)	1	2	3	4
N_t	500	6190	232284	1000000
N_a	0	852	183960	994043
A%	0	13.7%	79.2%	99.4%

仿真结果分析：从译码性能上看，SC-SCL 译码算法的性能要远远优于 SC 译码算法，这说明冻结比特差异度是能较准确的衡量译码结果的正确性的。比如当码长 $N = 512$ ，码率 $R = 0.5$ ，误比特率为 10^{-5} 时，SC-SCL 译码相较 SC 译码算法，能够获得约 0.8 dB 的编码增益。

从复杂度的角度分析：SC-SCL 译码算法的性能与 SCL 译码算法性能相当，但从表 5-7，表 5-8，表 5-9 的数据可以看出，大部分译码过程中，SC-SCL 算法都只进行了 SC 的译码操作，尤其是在高信噪比的条件下，SC-SCL 译码算法中只进行 SC 译码次数的比重达到 99% 以上，与 SCL 译码算法比较而言，极大地降低了译码复杂度，且得到冻结比特差异度值的过程本身也不会增加额外的复杂度。

5.4 本章小结

利用冻结比特是双发都已知的先验信息，本章提出了一种新的译码可靠性测度——冻结比特差异度 δ 。该参数值的获取本身非常简单，只需要在特定的位置，即 Q 集合中，根据虚拟子信道的 LLR 值进行硬判决并判断与冻结比特的设定值是否相符即可。与 CRC 校验和距离参数不同，冻结比特差异度不需要再进行额外的编码译码操作，更加简单明了。冻结比特差异度衡量译码结果的准确度取决于 Q 集合的选取，所以在本章中针对 Q 集合的选择提出了两种选择算法：巴氏参数法和 scaling exponent 选择法，并针对巴氏参数提出了计算巴氏参数的简化算法，以便快速获取 Q 集合。 δ 能够较为准确地衡量译码结果的正确性，基于 δ ，本章提出了一种自适应 SC-SCL 算法，该算法与 SC 译码算法相较，译码性能有较大提升，且与 SCL 译码算法的译码性能持平；但就复杂度而言，译码复杂度又远小于 SCL 译码算法，是一种兼顾性能与复杂度的译码算法。

6 结论

极化码源于信道极化，是目前唯一在理论上已证明可达到 BSC 信道容量的编码方案。对于极化码的研究方向主要分为三类：对于虚拟子信道的可靠性衡量，编码方式以及译码算法的研究。本文主要的研究方向是属于译码算法研究的一个子方向：对于译码结果进行可靠性估计，从而进一步提升译码性能。

SC 译码算法是基于极化码结构提出的一种译码算法，它充分利用了极化码的特性，是极化码最核心的一种译码方式。但 SC 译码算法存在一个问题，即无法通过译码结果本身对译码正确性进行判断。也就是说：SC 译码是“一锤子买卖”，译完最后一个信息比特即宣告译码过程结束，直接输出最后的译码结果，这样带来的问题是无法发现译码错误，也就无法对错误比特进行进一步处理，导致性能不理想。所以我们需要寻找译码可靠性测度。

首先，本文研究了最常见的一种译码可靠性测度：CRC。CRC 校验是一种比较常用的错误校验方式，在极化码研究中主要应用在单比特翻转译码算法和 CA-SCL 译码算法中用于提升译码性能。本文在 CA-SCL 译码算法的基础上，提出了基于关键集合的部分 CRC 辅助的 SCL (CS-PCA-SCL) 译码算法。在 CRC 校验位长度一致的情况下，CS-PCA-SCL 算法只针对关键集合中的元素进行校验，减少了监督的位数，进一步提高了 CRC 校验的准确性，从而提升了译码性能。从仿真结果看出，在低信噪比下，两种译码算法的 BER 基本相等；在高信噪比下，CS-PCA-SCL 算法的译码性能明显要优于 CA-SCL 算法。

其次，本文讨论了基于距离的译码可靠性测度：距离参数。通过公式推导得出，距离参数值越小，表示译码路径越可靠，通过距离参数选出来的最优码字符合最大似然译码规则。本文将距离参数应用在极化码译码算法中，提出了基于距离参数的单比特翻转 (D -SCF) 译码算法。通过用距离参数来代替 CRC 对译码结果进行衡量，消除了 CRC 校验位占据可靠信道的影响，从而提升译码性能，且随码长变长，译码增益更加明显。

最后，本文基于信道极化特性，提出一种新的译码可靠性测度：冻结比特差异度。冻结比特差异度是利用“冻结比特是收发双发都已知”的先验信息，统计译码时 Q 集合内的冻结比特的硬判决结果与冻结比特设定值不一致的个数而得到。针对 Q 集合，本文提出两种选择算法：巴氏参数选择法和 scaling exponent 选择法，两种算法都能较准确的选择 Q 集合，并且还提出了巴氏参数的简便算法，精简计算虚拟子信道的巴氏参数值的计算复杂度，加快寻找 Q 集合。对比两种选择算法的仿真结果，在高信噪比的情况下，两种算法的性能差异并不是很大，但在低信噪比的条件下

下，基于 scaling exponent 的选择算法的精度要高一些。

冻结比特差异度可以用来衡量译码结果的可靠性，于是本文提出了基于冻结比特差异度的自适应 SC-SCL 译码算法：在 SC 译码结束之后通过冻结比特差异度判断此次译码结果的可靠性，若不可靠，则采用 SCL 译码算法继续译码。仿真结果表明，SC-SCL 译码算法大幅度提高了 SC 译码算法的性能，且性能与 SCL 译码算法不相上下；但译码复杂度又远远低于 SCL 译码算法，分析仿真数据后发现大部分译码过程中，SC-SCL 算法都只进行了 SC 的译码操作，尤其是在高信噪比的条件下，SC-SCL 译码算法中只进行 SC 译码次数的比重达到 99% 以上。

本文研究的这三种译码可靠性测度各有利弊，每个参数都有其适合使用的场景。就参数计算复杂度而言，冻结比特差异度是最低的，不需要进行额外的运算，但冻结比特差异度对译码结果的衡量准确度比不上 CRC 和距离参数，在某些场景下并不适用。所以在不同的译码算法下需要根据具体情况选用不同的参数来衡量译码结果，以期进一步提高译码性能。

关于译码结果的衡量参数还有许多可研究的空间。比如：由于极化码的译码特性，在计算距离参数的时候总是需要进行重编码的过程，复杂度较高，如何快速的获取每一条译码路径的距离参数是一个未解决的问题。此外，是否可以找到更加普遍适用的译码结果的衡量参数，既不用增加额外的计算，又可以非常准确地判断译码结果的可靠性，这样一来，就可以在不增加复杂度的情况下大幅度提高极化码的译码性能，拓宽极化码的应用范围。

总而言之，相较于其他信道编码技术，极化码是一种非常年轻的编码方案。关于极化码的很多特性还有待人们去探索，进一步地提高极化码的性能将是一件非常有趣的研究工作，希望能在未来看到更多有关极化码的研究成果出现。

参考文献

- [1] 王键. 5G 信道编码技术相关分析[J]. 数字通信世界, 2019, 171(03):47+81.
- [2] Shannon C E. A mathematical theory of communication[J]. The Bell System Technical Journal 1948, 27(4):379-423.
- [3] Huang P C, CC Chang, Li Y H, et al. Efficient QR Code Secret Embedding Mechanism Based on Hamming Code[J]. IEEE Access, 2020, PP(99):1-1.
- [4] 张永光. RM 码特征分析[J]. 微电子学与计算机, 2020, 432(05):43-46.
- [5] Rao P, Babshet P, Babu R A, et al. Encoder and Adaptive Decoder for a (15,6,2) DEC-TED BCH Code[C] // 2020 IEEE 17th India Council International Conference. 2020: 1-6.
- [6] 蒋飘蓬, 刘杰, 马超. 基于软判决的(n,1,m)卷积码盲识别[J]. 舰船电子工程, 2018, 286(04):45-48.
- [7] 殷璐. Simulation Analysis of Turbo Code with CRC-check Termination[J]. 通信技术, 2017, 050(007):1371-1374.
- [8] Zhang B, Cosman P, Milstein L. Energy Optimization for Hybrid ARQ With Turbo Coding: Rate Adaptation and Allocation[J]. IEEE Transactions on Vehicular Technology, 2020, PP(99):1-1.
- [9] Thi H P, The C D, Xuan N P, et al. Simplified Variable Node Unit Architecture for Nonbinary LDPC Decoder[C] // 2019 IEEE Asia Pacific Conference on Circuits and Systems. 2020:213-216
- [10] Wang C X, Lin C H. Improved Normalized Probabilistic Minimum Summation Algorithm for LDPC Decoding[C] // 2020 IEEE International Conference on Consumer Electronics. 2020:1-2.
- [11] Chung S-Y, Forney G D, Richardson T J, et al. On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit[J]. IEEE Communications Letters, 2001,5(2):58-60.
- [12] Arikan E. Channel polarization: A method for constructing capacity-achieving codes[C] // 2008 IEEE International Symposium on Information Theory. 2008:1173-1177.
- [13] 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.
- [14] Zhang B, Shen H, Yin B, et al. A 5G Trial of Polar Code[C] //2016 IEEE Globecom Workshops. 2016:1-6
- [15] 牛凯. "太极混一"——极化码原理及 5G 应用[J]. 中兴通讯技术, 2019,144(01):23-32+66.
- [16] Arikan E. Channel combining and splitting for cutoff rate improvement[J]. IEEE Transactions on Information Theory, 2005, 52(2):628-639.
- [17] Zhao S, Shi P, Wang B. Designs of Bhattacharyya Parameter in the Construction of Polar Codes[J]. 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing, 2011:1-4
- [18] Mori R, Tanaka T. Performance of polar codes with the construction using density evolution[J]. IEEE Communications Letters, 2009,13(7):519-521.
- [19] Wu D, Li Y, Sun Y. 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.

- [20] Tal I, Vardy A. How to Construct Polar Codes[J]. IEEE Transactions on Information Theory, 2011, 59(10):6562-6582.
- [21] He G, Belfiore J C, Land I, et al. Beta-expansion: A theoretical framework for fast and recursive construction of polar codes [C] //2017 IEEE Global Communications Conference. 2017:1-6
- [22] Arikan E, Telatar E. On the rate of channel polarization[C] //2009 IEEE International Symposium on Information Theory. 2009:1493-1495.
- [23] Korada S B, Sasoglu E, Urbanke R. Polar Codes: Characterization of Exponent, Bounds, and Constructions[J]. IEEE Transactions on Information Theory, 2010, 56(12):6253-6264.
- [24] Presman N, Shapira O, Litsyn S, et al. Binary Polarization Kernels from Code Decompositions[J]. IEEE Transactions on Information Theory, 2014, 61(5):2227-2239.
- [25] Lin H P, Lin S, Abdel-Ghaffar K. Linear and Nonlinear Binary Kernels of Polar Codes of Small Dimensions with Maximum Exponents[J]. IEEE Transactions on Information Theory, 2015, 61(10):5253-5270.
- [26] Lee M K, Yang K. The exponent of a polarizing matrix constructed from the Kronecker product[J]. Designs codes and cryptography, 2014,70(3):313-322.
- [27] Eslami A, Pishro-Nik H. A Practical Approach to Polar Codes[C] // 2011 IEEE International Symposium on Information Theory Proceedings. 2011:16-20
- [28] Kai N, Lin J, Kai C. Beyond turbo codes: Rate-compatible punctured polar codes[C] // 2013 IEEE International Conference on Communications. 2013: 3423-3427.
- [29] Wang R, Liu R. A Novel Puncturing Scheme for Polar Codes[J]. IEEE Communications Letters, 2014, 18(12):2081-2084.
- [30] Liang Z, Zhang Z, Wang X, et al. On the puncturing patterns for punctured polar codes[C] // 2014 IEEE International Symposium on Information Theory. 2014:121-125
- [31] Chandesris L, Savin V, Declercq D. On puncturing strategies for polar codes[C] // 2017 IEEE International Conference on Communications Workshops. 2017:766-771.
- [32] Shin D M, Lim S C, Yang K. Design of Length-Compatible Polar Codes Based on the Reduction of Polarizing Matrices[J]. IEEE Transactions on Communications, 2013, 61(7):2593-2599.
- [33] Miloslavskaya V. Shortened Polar Codes[J]. IEEE Transactions on Information Theory, 2015, 61(9):4852-4865.
- [34] Sasoglu E, Telatar E, Arikan E. Polarization for arbitrary discrete memoryless channels[C] // 2009 IEEE Information Theory Workshop. 2009:144-148.
- [35] Mori R, Tanaka T. Channel Polarization on q-ary Discrete Memoryless Channels by Arbitrary Kernels[J]. 2010 IEEE International Symposium on Information Theory, 2010:894-898.
- [36] Mori R, Tanaka T. Non-Binary Polar Codes using Reed-Solomon Codes and Algebraic Geometry Codes[C] // 2010 IEEE Information Theory Workshop. 2010:1-5.
- [37] Presman N, Shapira O, Litsyn S. Polar Codes with Mixed-Kernels[J]. 2011 IEEE International Symposium on Information Theory Proceedings, 2011:6-10.
- [38] Presman N, Shapira O, Litsyn S. Mixed-Kernels Constructions of Polar Codes[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(2):239-253.
- [39] Ao R, Bo Y, Wang Y. Design of high-speed low-power polar BP decoder using emerging technologies[C] //2016 29th IEEE International System-on-Chip Conference. 2017:312-316.
- [40] 董云南. Polar码的BP译码算法的改进[D].西安: 西安电子科技大学, 2020.

- [41] Zhang Y, Zhang Q, Pan X, et al. A simplified belief propagation decoder for polar codes[C] // 2014 IEEE International Wireless Symposium. 2014:1-4.
- [42] Niu K, Chen K, Lin J. Low-Complexity Sphere Decoding of Polar Codes based on Optimum Path Metric[J]. IEEE Communications Letters, 2013,18(2):332-335.
- [43] Guo J, Albert G. Efficient sphere decoding of polar codes[C] // IEEE International Symposium on Information Theory. 2015:236-240.
- [44] 宋倩茹. 基于最优化理论的Polar码译码算法研究[D].西安: 西安电子科技大学, 2020.
- [45] Taranalli V, Siegel P H. Adaptive Linear Programming Decoding of Polar Codes[J]. 2014 IEEE International Symposium on Information Theory, 2014: 2982-2986.
- [46] Tal I, Vardy A. List Decoding of Polar Codes[J]. IEEE Transactions on Information Theory, 2015, 61(5):2213-2226.
- [47] Niu K, Chen K. CRC-Aided Decoding of Polar Codes[J]. IEEE Communications Letters, 2012, 16(10):1668-1671.
- [48] Yuan F, Tian B. Double-Parity-Check CA-SCL Encoding and Decoding for Polar Codes[C] // 2018 14th IEEE International Conference on Signal Processing. 2018:747-751.
- [49] Zhou H, Zhang C, Song W, et al. Segmented CRC-Aided SC List Polar Decoding[C] // 2016 IEEE 83rd Vehicular Technology Conference. 2016:1-5.
- [50] Zhang Q, Liu A, Pan X, et al. CRC Code Design for List Decoding of Polar Codes[J]. IEEE Communications Letters, 2017,21(6):1229-1232.
- [51] Niu K, Chen K. Stack decoding of polar codes[J]. Electronics Letters, 2012, 48(12):695-697.
- [52] Kai C, Niu K, Lin J R. Improved Successive Cancellation Decoding of Polar Codes[J]. Electronics Letters, 2012, 61(8):3100-3107.
- [53] Afisiadis O, Balatsoukas-Stimming A, Burg A P. A Low-Complexity Improved Successive Cancellation Decoder for Polar Codes[C] //2014 48th Asilomar Conference on Signals.2014:2116-2120.
- [54] 白慧卿. 基于极化码的物理层安全编码技术研究[D].郑州: 战略支援部队信息工程大学,2018.
- [55] 黄开枝,万政,楼洋明,肖帅芳,许晓明. 基于极化码的无协商密钥物理层安全传输方案[J].电子与信息学报,2020,42(12):2946-2952.
- [56] Mishra A, Sharma K, De A. Quality Image Transmission through AWGN Channel using Polar Codes[J]. International Journal of Computer Science and Telecommunications,2014,5(1).
- [57] Zhao S, Shi P, Wang B. Polar codes and its application in speech communication[C] //2011 International Conference on Wireless Communications and Signal Processing. 2011:1-4.
- [58] Leroux C, Tal I, Vardy A, et al. Hardware architectures for successive cancellation decoding of polar codes[C] //2011 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE. 2011:1665-1668.
- [59] Abbe E, Telatar E. Polar codes for the m-user multiple access channel and matroids[J]. IEEE Transactions on Information Theory, 2012,58(8): 5437-5448.
- [60] Korada S B, Urbanke R. Polar Codes are Optimal for Lossy Source Coding[J]. IEEE Transactions on Information Theory, 2010, 56(4):1751-1768.
- [61] Zhang Z, Qin K, Liang Z, et al. Progressive Bit-Flipping Decoding of Polar Codes over Layered Critical Sets[C] // 2017 IEEE Global Communications Conference. IEEE. 2017:1-6.

- [62] 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.
- [63] Chase D. Class of algorithms for decoding block codes with channel measurement information[J]. IEEE Transactions on Information Theory, 1972, 18(1):170-182.
- [64] 李海龙. 基于Chase算法的双路TPC编译码技术研究[D]. 哈尔滨: 哈尔滨工业大学,2015.
- [65] Korada S B, Montanari A, Telatar E, et al. An Empirical Scaling Law for Polar Codes[C] //2010 IEEE International Symposium on Information Theory. 2010:884-888.
- [66] Ryan W E, Shu L. 信道编码: 经典与现代[M]. 白宝明, 马啸, 译. 北京: 电子工业出版社, 2009

作者简历及攻读学位期间取得的研究成果

一、作者简历

2014年9月至2018年6月	燕山大学	电子信息工程	学士学位
2018年9月至2021年6月	北京交通大学	通信与信息系统	硕士学位

二、发表国际会议论文

[1] Xiong J, Zhang L. Simplified Calculation of Bhattacharyya Parameters in Polar Codes[C] // 2020 IEEE 14th International Conference on Anti-counterfeiting, Security, and Identification (ASID), 2020: 169-173.

三、参与科研项目

[1] 国家自然科学基金面上项目，61872031，基于熵理论的信息匹配网络测量与建模

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：

签字日期：

年 月 日

学位论文数据集

关键词*	密级*	中图分类号*	UDC	论文资助
极化码; 译码可靠性测度; CRC; 距离参数; 冻结比特差异度	公开			NSFC: 61872031
学位授予单位名称*		学位授予单位代码*	学位类别*	学位级别*
北京交通大学		10004	工学	硕士
论文题名*		并列题名		论文语种*
极化码译码可靠性测度及应用				汉语
作者姓名*	熊佳慧	学号*	18120151	
培养单位名称*	培养单位代码*	培养单位地址	邮编	
北京交通大学	10004	北京市海淀区西直门外上园村 3 号	100044	
学科专业*	研究方向*	学制*	学位授予年*	
通信与信息系统	信道编码	3 年	2021 年	
论文提交日期*	2021 年 5 月			
导师姓名*	张立军	职称*	副教授	
评阅人	答辩委员会主席*	答辩委员会成员		
	郭宇春	赵永祥 李纯喜 陈一帅 郑宏云		
电子版论文提交格式 文本 (√) 图像 () 视频 () 音频 () 多媒体 () 其他 () 推荐格式: application/msword; application/pdf				
电子版论文出版 (发布) 者	电子版论文出版 (发布) 地		权限声明	
论文总页数*	69			
共 33 项, 其中带*为必填数据, 为 22 项。				