极化码的编译码算法研究

第1章：绪论

* 1. 研究背景与意义

1948年10月， Claude Shannon（克劳德.香农）在他的经典论文“A Mathematical Theory of Communication”（通信的数学理论）[1]的引言部分中写道：

“通信的基本问题就是在某一点精确或者近似地再生另一点选择的信息。”

为了解决这个问题，他在该论文中系统的阐述了信息论的概念与方法，奠定了信息论的理论基础。此后，在信息论理论框架的指导下，保密编码，信源编码和信道编码三个领域平行发展，编码的新方法不断出现，编码性能越来越好。

香农在文献[1]中还提出了计算信道容量的香农公式：

(bit/s) (1)

香农公式揭示了信道无差错传输的条件：当发送端以bit/s的信息速率发送二进制数据时，一定存在某种高效的编码方式，可以实现无差错传输；当发送端以bit/s的速率发送二进制数据时，任何编码系统都无法实现无差错的传输。

香农公式给出了理想通信系统的标准：能达到极限信息速率并且差错率为零。但是，香农并没有给出实现理想通信系统的具体方式。在香农定理的指导下，科研工作者们一直致力于寻找一种能够达到信道极限而且复杂度也较低的编码方法，因此也涌现出了一批编译码方法。汉明码（Hamming code）[2]，格雷码（Gary code）[3]，里德穆勒码（Reed-muller code, RM）[4;5]等都是早期比较有代表性的编码方式。由于上述编码都是采用硬判决方式，并且只适用于短码，因此无法满足实际通信系统的需求。随着编码理论的飞速发展，采用软判决的LDPC码[6]以及Turbo码[7]相继问世，并且成为主流的编码方案，广泛的应用于经典通信系统中，例如：WCDMA，TD-SCDMA，CDMA2000等。这两种编码方式虽然极大的提升了误码性能，但是仍然没有达到信道极限，并且具有较高的复杂度。

在当前热门的第五代移动通信（5G）相关技术里，高效的信道编码方案的选取依然是其核心问题。LDPC码在2016年10月被采纳为5G增强型移动宽带（eMBB）场景下数据信道的长码编码方案。随后的2016年11月，在3GPP RAN1 #87会议上，由E. Arikan提出的极化码（Polar code）[7]被当选为5G eMBB场景下的控制信道编码方案。LDPC码和极化码，是信道编码领域两大热门的研究方向。因此研究极化码的高性能编解码方案，具有很高的实际意义。

* 1. 极化码的发展历史与现状

由E. Arikan提出的极化码[7]已经被证明是一种能够在二进制离散无记忆信道（Binary-input discrete memoryless channels, B-DMCs）条件下，达到信道极限的编码方案。串行抵消（Successive cancellation, SC）译码也在文献[7] 中被提出，该种译码算法具有很低的译码复杂度，仅为。然而，在中等码长条件下，由于比特信道的不充分极化现象， SC译码的误码纠错性能令人不甚满意。为了获得在有限码长条件下，极化码更好的性能，文献[8;9;10]提出了串行抵消列表（Successive cancellation list, SCL）译码算法。文献[8;9;10]还展示了添加循环冗余检测（Cyclic redundancy click, CRC）的SCL译码算法（用CA-SCL表示），其误码性能可以与LDPC码匹配。文献[11;12;13;14]介绍了SC译码的硬件实现。SCL译码的硬件实现可以参考文献[15;16;17;18]。通过极化码的硬件实现结果可以看出，CA-SCL译码的优良性能是以相对高的译码复杂度和低的吞吐率为代价的。

置信传播（Belief propagation, BP）译码[19;20] 是另一种基于极化码因子图的译码算法。BP 译码的性能优于SC译码，并且是并行译码。但是BP译码的BER性能仍然次于SCL译码。

SCF (Successive cancellation flip) 译码算法[21] 是改进SC译码算法的另一种方式。它拥有比SCL更低的硬件实现复杂度，并且能够达到列表数为2时的SCL译码算法的性能，但是牺牲了译码延迟。基于对SC译码算法的第一个错误估计的比特的分布情况，文献[22] 提出了两种改进的SCF译码算法。SCF译码算法的核心思路是，在SC译码算法出现错误时，找到其第一个错误估计的信息比特，并将其翻转（译码值取反），然后继续译码。文献[21; 22] 还给出了SC译码算法的错误传播规律，因此，如果能保证在SC译码过程中，最容易错误估计的信息比特的正确性，那么SC译码算法的性能将大幅度提升。

受SCF译码算法性能的影响，许多科研工作者开始研究如何保证在SC译码算法过程中，对应信道质量最差的信息比特的正确估计。文献[23;24;25] 提出了带有共享信息比特的极化码编码方法。其中，文献[23] 提出了（Star polar subcodes, SPS）, 该编码方法将极化码编码为星形结构，各个码块之间在一些信息比特的重叠，从而达到共享一部分信息比特的效果。文献[24] 提出的部分信息位耦合（Partially information coupled, PIC）极化码采用了系统极化码（Systematic polar codes, SPC）的编码方式，使得经过编码的码字，其每个信息块的首尾分别与相邻信息块共享相同的信息比特。PIC码还使用了删余（puncture） 技术，以减小码率的损失。文献[25] 提出了（Inter-frame polar codes, IPC）该算法在每个编码的信息块中插入了一定数量的动态冻结位，即每个信息块中，最可靠的冻结位比特（Most reliable frozen bits, MRFB）与前一个信息块中最不可靠的非冻结比特（Most unreliable un-frozen bits, MUUB）保持一致。上述几种极化码的编码结构均可以实现当某个信息块译码失败时，可以利用相邻的，译码成功的信息块中的正确共享信息比特，再重新译码一次。通过这种方式可以实现极化码性能的提升。

目前的已有的带有共享信息比特的极化码编码方法，存在着以下问题：

* 译码错误的信息块，其纠正条件难满足：在IPC的译码过程中，当某个信息块没有通过CRC校验时，则只有在其相邻的两个信息块均译码成功的条件下，该信息块才能获取正确的MRFB和MUUB，并进行新一轮的译码。并且，当某个信息块译码错误时，在下一个信息块的译码过程中，其MRFB会被当作信息比特来译码，导致了更大的误组率（Packet error rate, PER）。因此降低了错误信息块第二轮译码发生的概率。PIC码采用了系统码编码，但是其译码的思路和IPC是一致的。
* 硬件实现的复杂度较高。
  1. 本文主要研究工作与结构安排

本文提出了一种极化码的新型编译码方案：在两块紧密相连的编码块中，共享一定数量的信息比特。系统的输入比特流被划分为奇信息比特和偶信息比特。在编码过程中，一组奇偶信息块共享一小块信息比特，并称之为互信息比特（Mutual information bits, MIB）。每个信息块还包含了CRC比特。在译码阶段，当一对信息块中只有一组译码失败时，译码正确的信息块可以给译码错误的信息块提供关于MIB的硬判决信息。合理地设计MIB的位置，可以使得译码错误的信息块开始新一轮的译码，并且被纠正过来。

由于是两个紧密相连的信息块共享MIB，所以这种极化码的编译码方案看起来就像是在编码阶段存在一些记忆存储单元。因此，这种方案被命名为有记忆的极化码（Polar codes with memory, PCM），用以区分传统的极化码编译码方案。另外，PCM还可以直接拓展到个信息块的场景。基于此，本文还提出了一种普适型PCM（General PCM, G-PCM）方案，与直接拓展PCM的方案相比，该方案具有更小的码率损失，并且能维持相似的PER。PCM的底层译码器可以是SC，BP或者SCL译码器，其译码复杂度极为底层译码器的复杂度。

为了描述方便，我们用PCM-SC-来表示拥有个共享MIB的信息块，且使用SC译码器的PCM方案。同理，PCM-BP-和PCM-SCL-分别代表使用BP以及SCL译码器的PCM方案。

本文的主要贡献如下：

1. 提出了PCM编译码方案，包括PCM的系统模型，编译码策略，误码性能分析。仿真结果表明，在PER级别下，PCM-SC-2的PER性能与列表数为2的SCL译码算法的性能相比，仅仅相差0.2 dB。PCM-BP-2可以达到列表数为2的SCL相同的性能。另外，PCM-SCL-2的在使用列表数为的SCL译码器时，其误码性能可以与列表数为的SCL的性能相匹配。
2. 拓展PCM至的场景，并且提出了G-PCM方案。与PCM的直接拓展方案相比，该方案可以实现更小的码率损失。PCM-SC-3的仿真结果表明，G-PCM方案没有损失误码性能。
3. 提出了两种实现PCM的硬件结构，即顺序（In-serial, IS）结构和低延迟交织（Low-latency interleaved, LLI）结构。硬件实现的结果表明，在码长的条件下，与文献[17]中的SCL译码器相比，PCM-SC-2的LLI硬件结构具有较低的译码延迟以及更高的吞吐率。

文章剩余的结构安排如下：

第2章简单介绍极化码的基本原理，编码方法以及常见的译码算法。第3章系统的介绍PCM方案。具体来说，第3章第1小节介绍PCM的编码方案；第2小节介绍PCM的译码策略；第3小节介绍MIB的最佳选择；第4小节分析了PCM的误码性能；采用BP和SCL译码的PCM出现在第5小节；在第6小节，我们还将PCM于Turbo码做了对比。第4章是关于PCM的拓展。第5章提供了关于PCM的仿真结果。在第6章，我们介绍了PCM的两种硬件实现。最后一章是总结部分。

第2章：极化码的相关工作

* 1. 引言部分

Arikan在文献[7]中详细的介绍了极化码的极化原理。用来表示一个一般类型的B-DMC，其中表示输入字符集，表示输出字符集，则信道转移概率表示为。输入字符集满足，输出字符集和转移概率可以是任意的。用表示一种对的次使用的信道；因此有：，并且转移概率为。

对于任意给定的B-DMC信道，存在两个性能参数：对称容量（Symmetric capacity）

(1)

和巴氏参数（Bhattacharyya parameter）

. (2)

这两个参数可以用来衡量信道传输的速率和可靠性。是在信息在信道中可靠传输所能达到的最大信息速率。是当信道仅传输二进制比特时，其采用最大似然（Maximum-likelihood, ML）译码算法的误码率上限。很显然，与的取值范围均在[0, 1]之间。

定义表示一个长度为N的向量。在给定且集合的条件下，用来表示子向量（）。表示关于的补集。此外，表示集合中的元素个数。

* 1. 信道极化原理

信道极化是一个利用给定的B-DMC信道的个独立复制信道生成个极化信道的过程。信道极化带来的影响是，当非常大时，会趋于0或者1。信道极化分为信道联合（Channel combining）阶段和信道分裂（Channel splitting）阶段。

* + 1. 信道联合

信道联合是一个将给定的B-DMC信道的重独立复制信道以一种连续迭代的方式组合成一个信道向量，其中。信道联合开始于，此时只有信道本身，因此。当时，两个独立的结合生成，其转移概率为

. (3)

图2.1展示了联合信道形成的过程。



图2.1：联合信道。

图2.2显示了接下来的信道联合迭代过程。两个独立的结合，形成了联合信道，其转移概率为。



图2.2：联合信道与信道和的关系。

图2.2中的是一个转置操作，将输入映射成。源信息比特经过联合信道，由信道输出码字，其映射关系可以表示为，其中。由此可得。

图2.3为信道联合的一般形式示意图，其中两个独立复制的信道联合生成了信道。信道的输入比特首先被转换成，并且对任意，均有，。图中所示的被称为反转变换操作，它将输入信号转换成，然后成为信道的输入信号。



图2.3：联合信道的构造。

在图中我们还可以看出，映射是一个在2进制数范围内的线性变换。同样可以推导出，联合信道的输入和底层信道的输入信号也满足线性变换关系，可以用矩阵来表示，即。被成为阶生成矩阵（Generator matrix）。对于所有的，，信道到的转移概率为

. (4)

生成矩阵满足关系式，其中是一个实现比特反转（Bit-reversal）的转置矩阵，。

* + 1. 信道分裂

在信道联合阶段，信道经过构造生成联合信道，信道极化的下一个阶段，信道分裂，即将生成的分裂，重新生成个并列的比特信道，。定义其转移概率为

(5) 其中表示信道的输出，表示其输入。

* + 1. 信道极化

信道极化定理：对于任意给定的B-DMC类型的，其极化信道满足，当趋向于无穷大时，其对应的信道对称容量趋向于1或者0.

该定理的证明已经在文献[7]中给出。图2.4展示了当信道为BEC信道，且擦出概率为时，信道极化的效果。可以通过下面两个迭代公式来计算：

(6)

(7)

其中。



图2.4：BEC信道极化后的对称容量，擦出概率为，。

从图2.4也可以看出，在非常小的时候趋向于0，在非常大的时候趋向于1。信道极化后，可以选择一个合理的值，来挑选出质量较好的信道，传输信息比特。

* 1. 极化码的编码

极化码的编码即为利用信道极化的现象，构造出一种可以达到对称信道容量的编码过程。极化码编码的基本思想是，对于所有的独立的极化信道，只在那些信道参数接近0（即趋于1）的信道中传输数据。

对于任意码长，，且，极化码的编码公式可以表示为

(8)

为极化码的生成矩阵，在前面已经简单介绍过。假设集合为集合的任意子集，公式（8）可以重写为

（9）

其中表示抽取集合中的元素对应的行向量形成的子矩阵。

通过公式（9）可以得到，对于确定的集合和向量，存在一个由源信息比特到码字的映射关系，该映射关系其实就是一个关于的陪集码，它由固定的向量决定。我们将这种类型的陪集码称为陪集码，可以用参数向量来表示，其中即为集合的元素个数。此为，比率被称之为码率（Code rate）。定义集合为信息位集合，为冻结比特向量。以下是一个简单的例子：

码的编码公式可以表示为

其中源信息比特为，编码得到的码字为。

极化码是一种特殊的陪集码，其信息位集合由信道参数决定。由于现实中的码长不可能是无限长的，所以存在部分比特信道不完全极化现象。因此，极化码在编码时，需要根据信道参数在个比特信道中挑选出个最可靠的信道，传输信息比特。固定的冻结位则在剩余的比特信道中传输。

* 1. 常用译码算法

极化码的译码算法主要有SC译码算法，BP译码算法以及SCL译码算法。我们将分别介绍这三种常用的译码算法。

* + 1. SC译码

SC译码算法由Arikan在文献[7]中随着极化码一起提出。它是一种顺序译码的算法。对于每一个信息比特，其似然值（Likelihood ratio, LR）可以用下式计算

（10）

信息比特可以根据下式进行判决：

（11）

对于冻结比特，由于其在发送端和接收端都是已知的，所以当满足时，直接有。文献[7]中还提供了计算似然值的迭代公式：

（12）

（13）

* + 1. SCL译码

文献[8;9;10]提出了SCL译码算法，SCL译码器可以在极化码译码树中搜索正确的译码路径，并且允许同时追踪条译码的最佳路径。对于每一条译码路径，都存在一个名为路径度量（Path metric, PM）值的参数，用以衡量该路径的可靠程度。当PM值越小的时候，对应的译码路径正确率越高。SCL译码器从第一个判决比特开始，在每个信息比特的判决过程中，都会将当前保存的列表数翻倍，直至列表数已达到设置上限条。此时，SCL译码器会先将列表数翻倍为条，然后再对这条路径按照其PM值进行排序，并保留条PM值最小的路径。PM的初始值为0，通过下面的方程来计算：

（14）

其中为PM值对应的第几条路径，且，为当前的判决比特位置。为了计算方便，我们使用对数似然值（Log-likelihood ratio, LLR）来代替LR值，公式（14）中的即为信息比特的LLR值，它由下式给出

（15）

在最后的信息比特判决阶段，SCL译码器会找到PM值最小的路径对应的判决比特，作为译码器的输出。

图2.5显示了一个SCL译码器在码树中搜索路径并且根据PM值的大小最后得到最佳路径的译码示意图。其发送信号的参数向量为，源信息比特向量为。假设极化码的传输信道为AWGN信道，且接收端计算出的各个子信道的对数似然比为。由图可以看出，SCL译码器最终保留的路径，其译码输出结果为。需要注意对于此刻的译码，必须要做一次比特反转操作，才能得到正确的输出，这是由于极化码的在编码时是经过了比特反转的。



图2.5：SCL译码示例。

文献[10]还提出了添加CRC的SCL译码算法（CRC-aided SCL, CA-SCL）。在极化码的编码阶段，少量的CRC比特被添加到了信息位中。然后，在最后的判决输出阶段，SCL译码器将保留的条路径按照PM值递增的顺序排列，再依次进行CRC校验。第一个通过CRC校验的路径将被作为结果输出。当所有的路径都没有通过CRC校验时，SCL译码器输出PM值最小的那条路径。

* + 1. BP译码算法

在极化码的BP译码过程中，消息的传递是基于文献[19;20;25]的极化码因子图的。图2.6显示了极化码因子图中，消息传递的基本模块，它也被称为流程元素（Process element, PE）。对于一个码长为的极化码，其因子图可以划分为阶（从第0阶开始），并且每一阶包含个PE。所有的消息都是用LLR值来表示，并且被划分为两类：从左到右传递和从右到左传递。如图2.6所示，我们用，来表示一个经过节点的，从右边到左边传递的消息；同样的，我们用来表示一个经过节点且从左向右传递的消息。



图2.6：BP译码器的基本PE。

对于每个节点，其消息传递可以通过下列迭代公式进行计算：

（16）

（17）

其中 ，是一个缩放因子。

图2.7是一个码长的BP译码因子图。图中的和分别表示与的初始值向量，它们可以用以下公式来计算：

（18）

（19）

初始化LLR值后，BP译码器对于每一个节点，根据公式（16）（17），按照先从右向左的再从左向右的顺序，进行LLR的更新迭代。当迭代次数达到预先设定的最大次数时，根据最左侧的LLR值，通过下式输出判决结果：

（20）



图2.7：条件下的极化码BP译码示意图。

第3章：有记忆的极化码

* 1. 引言

本章我们开始介绍有记忆的极化码（PCM），先介绍PCM的编码原理，然后介绍其译码策略。在PCM系统中，存在一个很重要的问题，即互信息比特（MIB）的位置选择问题，它直接影响了PCM的译码性能。我们在本章提出了该问题的解决办法，并加以证明。最后，我们还分析了PCM的误码性能，并将PCM与Turbo作了比较。

* 1. 编码原理
  2. 译码过程
  3. 互信息位选择
  4. 误码率分析
  5. 与Turbo码（LDPC码）的对比

第４章：多信息块的有记忆的极化码

* 1. 有记忆的极化码的直接扩展
  2. 一种通用的有记忆的极化码

第５章：仿真结果与分析

第６章：有记忆的极化码的硬件实现

* 1. 两种有记忆的极化码的译码器结构
  2. 硬件实现结果

第７章：总结与展望

* 1. 全文主要工作总结
  2. 下一步研究方向