第41卷 第4期 电 子 与 信 息 学 报 Vol. 41No. 4
2019年4月 Journal of Electronics & Information Technology Apr. 2019

基于预译码的极化码最大似然简化连续消除译码算法

刘建航* 何怡静 李世宝 卢丽金 邓云强

(中国石油大学(华东)计算机与通信工程学院 青岛 266580)

摘 要：针对极化码译码串行输出造成较大译码时延的问题，该文提出一种基于预译码的最大似然简化连续消除译码算法。首先对译码树节点存储的似然值进行符号提取并分组处理，得到符号向量组；然后比较符号向量组与该节点的某些信息位的取值情况，发现向量组中储存的正负符号分布规律与该节点的中间信息位的取值具有一一对应的关系；在此基础上对组合码中间的1～2 bit进行预译码；最后结合最大似然译码方法估计组合码中的剩余信息位，从而得到最终的译码结果。仿真结果表明：在不影响误码性能的情况下，所提算法与已有的算法相比可有效降低译码时延。

关键词：极化码；简化连续删除译码算法；最大似然译码；预译码

中图分类号：TN92 **文献标识码**：A **文章编号**：1009-5896(2019)04-0959-08

DOI: 10.11999/JEIT180324

Pre-decoding Based Maximum-likelihood Simplified Successive-cancellation Decoding of Polar Codes

LIU Jianhang HE Yijing LI Shibao LU Lijin DENG Yunqiang

(*College of Computer and Communication Engineering, China University of Petroleum* (*East China*), *Qingdao* 266580, *China*)

Abstract: To solve the long decoding latency caused by the serial nature of the decoding of polar codes, a pre-decoding based maximum-likelihood simplified successive-cancellation decoding algorithm is proposed. First, the signs of the likelihood values stored in the decoding tree nodes are extracted and grouped to obtain symbol vectors. Then comparing the symbol vectors and the values of some information bits, the distribution rules are found that positive and negative values stored in the vectors are one-to-one corresponding to the value of middle information bits of the node. Based on the above analysis, one or two bits in the middle of the constituent code are pre-decoded. Finally, the maximum likelihood decoding method is used to estimate the remaining information bits in the constituent code, and the final decoding results are obtained. Simulation results show that the proposed algorithm can effectively reduce the decoding delay compared with the existing algorithms without affecting the error performance.

Key words: Polar codes; Simplified successive-cancellation decoding; Maximum-likelihood decoding; Pre-decoding

1 引言

极化码是目前已知唯一的一种被严格证明达到信道容量的信道编码方法[1]，可以以较低的实现复杂度获得与最大自然译码相近的性能。因此，极化码可以应用到许多相关的理论研究领域，例如编调制技术[2]、NAND Flash[3]、窃听信道[4]以及物联网应用的短数据包场景[5]等。另外，极化码被作为5G移动通信中短码编码标准，在未来的通信发展中具有巨大的潜力。

然而用低复杂度的连续消除(Successive Cancellation, SC)译码算法[1]译码时，需要依次对每个比特译码，造成较大时延[6]。为了解决串行输出的译码时延问题，学者们提出了许多种译码方案，如球形译码[7,8]、并行译码[9–11]等。其中，Alamdar等人[9]提出了一种并行译码的译码方式，

收稿日期：2018-04-11；改回日期：2019-01-17；网络出版：2019-01-30
*通信作者： 刘建航 liujianhang@upc.edu.cn
基金项目：国家自然科学基金青年基金(61601519, 61872385)，中央高校基本科研业务费专项资金(18CX02134A, 18CX02137A, 18CX02133A)

Foundation Items: The National Natural Science Foundation of China (61601519, 61872385), The Fundamental Research Funds for the Central Universities(18CX02134A, 18CX02137A, 18CX02133A)

将极化码看作是多个组合码的级联，对都是信息比特的组合码(即rate-1组合码)直接进行最大似然(Maximum Likelihood, ML)判决，判决后的结果与SC译码结果一致。在此基础上，更多的结构特殊的组合码被找出，文献[10]提出快速简化连续删除(Fast Simplified Successive-Cancellation, Fast-SSC)译码算法，简化了SPC, REP和REP-SPC这3种组合码的译码过程；文献[11]又根据比特序列中固定位的不同提出了5种译码器来提升极化码译码速率。但是，以上方法都是基于启发式的，需要对每一种组合码使用不同的译码器译码，不适用于结构更为普通的组合码。对于更为普通的情况，文献[12–14]提出任意结构的组合码，只要满足一定条件同样可以用ML判决的方式直接译码，然而在信噪比较低的环境中，满足该条件的组合码占整个码字的比例较小。文献[15]提出的最大似然节点的简化连续消除(Simplified Successive-Cancellation with Maximum-Likelihood nodes, ML-SSC)译码算法是通过预计算未来码字的似然值，在估计当前组合码时选择适当的值作为译码输出。该方法对组合码的结构没有特殊的要求，并且不受信道条件的影响，可以很好地作为文献[10,11]的补充算法，进一步提升译码速率。不足之处是ML-SSC译码算法的译码效果会受到处理资源的约束，如果组合码较大，需要足够大的处理资源计算每个候选码字的似然值。

为了进一步降低译码时延，解决更多的普通组合码的简化译码问题，本文提出一种基于预译码的最大似然连续消除(Pre-Decoding based ML-SSC, PDM-SSC)译码算法，通过对组合码中间的1～2 bit进行预估计，然后再计算正在译码的组合码的最大似然码字，得到最终的译码结果。

2 极化码

2.1 极化码的构造

极化码是基于信道极化[1]来构建的，即在信道极化的基础上，一部分容量趋于1的信道传输信息比特，剩下的信道传输收发端都已知的固定比特。给定极化码长度为$N=2^n$，其中n为正整数，信息比特的个数为K，则码率R可表示为$R=K/N$。二进制信源序列$\boldsymbol{u}=(u_0,u_1,\cdots,u_{N-1})$包括信息比特子序列$\boldsymbol{u}_A$和固定比特子序列$\boldsymbol{u}_{A^{\mathrm{C}}}$，其中$A$为信息比特索引集$A\subseteq\{0,1,\cdots,N-1\}$，$A^{\mathrm{C}}$为$A$的补集，表示固定比特索引集。极化码编码后的序列可由式(1)运算得到

$$\boldsymbol{x}=\boldsymbol{u}\boldsymbol{B}_N\boldsymbol{F}^{\otimes n} \tag{1}$$

其中，$\boldsymbol{x}=(x_0,x_1,\cdots,x_{N-1})$，表示编码后的比特序列。$\boldsymbol{B}_N$为$N\times N$的排序矩阵，$\boldsymbol{F}^{\otimes n}$表示对矩阵$\boldsymbol{F}$进行$n$阶克罗内克积操作，且

$$\boldsymbol{F}=\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix} \tag{2}$$

由于极化码的生成矩阵$\boldsymbol{G}_N=\boldsymbol{B}_N\boldsymbol{F}^{\otimes n}$，则式(1)又可表示为

$$\boldsymbol{x}=\boldsymbol{u}\boldsymbol{G}_N \tag{3}$$

2.2 SC译码算法

定义信道输出向量$\boldsymbol{y}=(y_0,y_1,\cdots,y_{N-1})$，从信道接收到的对数似然比率(Log-Likelihood Ratio, LLR)为$(\lambda_0,\lambda_1,\cdots,\lambda_{N-1})$，其中

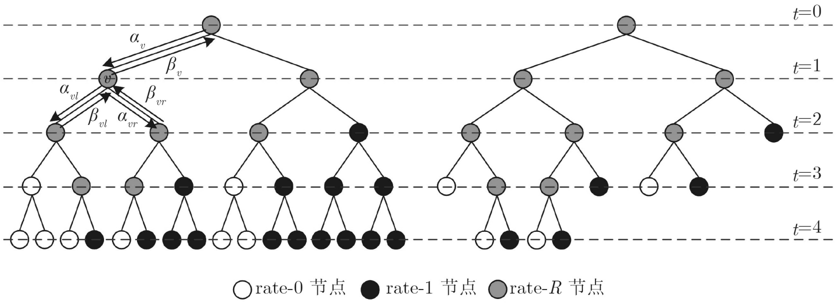
$$\lambda_i=\lg\frac{\Pr(y_i|x_i=0)}{\Pr(y_i|x_i=1)} \tag{4}$$

通过向译码树的根节点馈入LLR来初始化SC译码算法[9]。如图1(a)所示，节点v从其父节点接收到软信息向量$\boldsymbol{\alpha}_v$，经过计算将产生的码字$\boldsymbol{\beta}_v$传给其父节点。每当节点v被激活，根据式(5)得到该节点的$\boldsymbol{\alpha}_{vl}$并把它传递给左子节点v_l；直到从v_l中接收到码字$\boldsymbol{\beta}_{vl}$时，再由式(6)得到$\boldsymbol{\alpha}_{vr}$。

$$\boldsymbol{\alpha}_{vl}[i]=\boldsymbol{\alpha}_v[2i]\boxplus\boldsymbol{\alpha}_v[2i+1] \tag{5}$$

$$\boldsymbol{\alpha}_{vr}[i]=(1-2\boldsymbol{\beta}_{vl}[i])\,\boldsymbol{\alpha}_v[2i]+\boldsymbol{\alpha}_v[2i+1] \tag{6}$$

其中，$i=0,1,\cdots,2^{n-t-1}-1$。操作符“$\boxplus$”表示文献[15]中的最小和近似的方法



(a) SC 译码树 (b) SSC 译码树

图 1 SC及SSC译码算法对应的译码树

$$a \boxplus b = \mathrm{sign}\,(a)\,\mathrm{sign}\,(b)\min\left(|a|\,,|b|\right) \tag{7}$$

然后将$\boldsymbol{\alpha}_{vr}$传递给右子节点v_r，直到从v_r中接收到码字$\boldsymbol{\beta}_{vr}$再通过式(8)和式(9)得到码字$\boldsymbol{\beta}_v$

$$\boldsymbol{\beta}_v[2i+1] = \boldsymbol{\beta}_{vr}[i] \tag{8}$$

$$\boldsymbol{\beta}_v[2i] = \boldsymbol{\beta}_{vl}[i] \oplus \boldsymbol{\beta}_{vr}[i] \tag{9}$$

当节点v是叶子节点，可以直接将$\boldsymbol{\alpha}_v$输入二进制量化器中得到$\boldsymbol{\beta}_v$，即$\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v)$，再将$\boldsymbol{\beta}_v$传给它的父节点。$h(x)$表示二进制量化器，$x>0$时取值为0，$x<0$时取值为1，$x=0$时，则以1/2的概率取1和0。

2.3 SSC译码算法

SSC译码算法[9]将译码节点分为rate-1，rate-0和rate-R 3类，其中，rate-0节点表示该节点下所有子节点都对应固定比特；rate-1节点表示该节点下所有子节点都对应信息比特；rate-R节点下子节点中既有固定比特又有信息比特。然后简化rate-1节点的译码方法，通过式(10)和式(11)直接得到rate-1节点的译码比特，而不需要遍历其子树，如图1(b)。

$$\boldsymbol{\beta}_v = \mathrm{h}\left(\boldsymbol{\alpha}_v\right) \tag{10}$$

$$\left(\widehat{\boldsymbol{u}}\left[\min\mathcal{L}_v\right],\cdots,\widehat{\boldsymbol{u}}\left[\min\mathcal{L}_v\right]\right) = \boldsymbol{\beta}_v\boldsymbol{G}_{n-t} \tag{11}$$

其中，$\mathcal{L}_v$表示节点v下所有叶子节点的索引的集合；$\widehat{\boldsymbol{u}}$表示估计码字。

3　基于预译码的最大似然连续消除(PDM-SSC)译码算法

本节提出了新的两类节点，即L-REP节点和L-BiREP节点，针对这两类节点分析中间信息位传输的比特与节点的输入向量之间的对应关系，在此基础上确定预译码方案。再根据候选码枚举的方法，选择具有最大似然值的码字作为译码输出。

3.1 L-REP节点的预译码处理

假设T_n是深度为n的完全二叉树，包含2^n个叶子节点。已知节点v下共有2^{n-t}个叶子节点，对应组合码的长度为N_v，并且叶子节点的个数与组合码的长度相等。中间叶子节点的索引值为$(N_v-2)/2$，定义变量$\mathrm{mid0} = (N_v-2)/2$, $\mathrm{mid1} = (N_v-4)/2$。当节点$v$满足前mid1个叶子节点为固定节点，并且第mid0个叶子节点为信息节点，称节点v为L-REP节点。本小节将讨论通过对$\boldsymbol{\alpha}_v$进行处理，预估计L-REP节点的第mid0位比特的译码输出。

由2.1节可知，给定一个信道条件已知的传输信道，构建在该信道中传输的极化码前需要得到对应的生成矩阵$\boldsymbol{G}_N$。例如，为了构建$N=8$的极化码，需要先计算得到$\boldsymbol{G}_8$

$$\boldsymbol{G}_8 = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 1&0&0&0&1&0&0&0\\ 1&0&1&0&0&0&0&0\\ 1&0&1&0&1&0&1&0\\ 1&1&0&0&0&0&0&0\\ 1&1&0&0&1&1&0&0\\ 1&1&1&1&0&0&0&0\\ 1&1&1&1&1&1&1&1 \end{bmatrix}$$

分别用$\boldsymbol{g}_i$, $\boldsymbol{f}_i$表示生成矩阵$\boldsymbol{G}_N$和矩阵$\boldsymbol{F}^{\otimes n}$中的第$i$行行向量；定义连续因子$\mathrm{d}\left(\boldsymbol{x}\right)$为向量$\boldsymbol{x}$中连续1的最少个数，则$\boldsymbol{f}_i$中连续1的最少个数为$\mathrm{d}\left(\boldsymbol{f}_i\right)$, $\boldsymbol{g}_i$中连续1的最少个数为$\mathrm{d}\left(\boldsymbol{g}_i\right)$，如在$\boldsymbol{G}_8$中$\mathrm{d}\left(\boldsymbol{g}_3\right)=1$, $\mathrm{d}\left(\boldsymbol{g}_4\right)=2$。

引理1　$\forall N \geq 2$，且$\log_2 N \in \mathrm{N}^+$, $i \subset \{0,1,\cdots,N-1\}$，当$i=(N-2)/2$时，$\boldsymbol{g}_i = [1\ 0\ \cdots\ 1\ 0]$；当　$(N-2)/2 < i \leq N-1$　时　，$\mathrm{d}\left(\boldsymbol{g}_i\right) \geq 2$　且　$\log_2\mathrm{d}\left(\boldsymbol{g}_i\right) \in \mathrm{N}^+$　。

证明　假设矩阵$\boldsymbol{X}$, $\boldsymbol{Y}$均为N维矩阵，且有$\boldsymbol{Y} = \boldsymbol{B}_N\boldsymbol{X}$，由于$\boldsymbol{B}_N$是排序矩阵，矩阵$\boldsymbol{Y}$的第$(N-2)/2$行行向量等于矩阵$\boldsymbol{X}$中的第$N-2$行行向量。又因$\boldsymbol{G}_N = \boldsymbol{B}_N\boldsymbol{F}^{\otimes n}$, $\boldsymbol{G}_N$的第$(N-2)/2$行行向量等于$\boldsymbol{F}^{\otimes n}$的第$N-2$行行向量

$$\boldsymbol{g}_{(N-2)/2} = \boldsymbol{f}_{N-2} \tag{12}$$

已知克罗内积核$\boldsymbol{F} = \begin{bmatrix} 1 & 0; & 1 & 1 \end{bmatrix}$，接下来证明对于$N=2^k$，$k$为正整数，有

$$\boldsymbol{f}_{N-2} = \begin{bmatrix} 1 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{13}$$

当$k=1$时，$\boldsymbol{F} = [1\ 0;\ 1\ 1]$，则$\boldsymbol{f}_0 = [1\ 0]$，因此可证明$k=1$时命题成立。

假设$k=m$时，上述命题成立，即

$$\boldsymbol{f}_{2^m-2} = \begin{bmatrix} 1 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{14}$$

当　$k=m+1$　时　，$\boldsymbol{F}^{\otimes(m+1)} = \begin{bmatrix} \boldsymbol{F}^{\otimes m} & 0; & \boldsymbol{F}^{\otimes m} & \boldsymbol{F}^{\otimes m} \end{bmatrix}$　，因此

$$\boldsymbol{f}_{2^{m+1}-2} = \begin{bmatrix} \boldsymbol{f}_{2^m-2} & \boldsymbol{f}_{2^m-2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{15}$$

由上可知，$k=m+1$时，上述命题成立。

将式(13)代入式(12)中，得$\boldsymbol{g}_{N-2/2} = [1\ 0\ \cdots\ 1\ 0]$。根据克罗内积的运算规律可知，对于$\boldsymbol{F}^{\otimes n}$，有$\mathrm{d}\left(\boldsymbol{f}_{2i}\right)=1$且$\mathrm{d}\left(\boldsymbol{f}_{2i+1}\right) \geq 2$, $\log_2\mathrm{d}\left(\boldsymbol{f}_{2i+1}\right) \in \mathrm{N}^+$。设任意一个奇数$o \subset \{0,1,\cdots,N-1\}$，因为$o$的二进制最后一位一定为1，若将此奇数进行二进制转换，然后将每一位比特反序重排，得到新的二进制数，其最高位一定为1。根据以上规律，如果将矩阵$\boldsymbol{F}^{\otimes n}$进行反序重排得到矩阵$\boldsymbol{G}_N$, $\boldsymbol{F}^{\otimes n}$的奇数行行向量都排序到矩阵$\boldsymbol{G}_N$的后半部分中，即集合$\{\boldsymbol{f}_{2i+1}: i=0,1,\cdots,N/2-1\}$与集合$\{\boldsymbol{g}_j: j=N/2, N/2+1,\cdots,N-1\}$相同。由于$\mathrm{d}\left(\boldsymbol{f}_{2i+1}\right) \geq 2$，有

$\mathrm{d}\left(\boldsymbol{g}_j\right) \geq 2$ 且$\log_2 \mathrm{d}\left(\boldsymbol{g}_j\right) \in \mathrm{N}^+$，其中$j = \{N/2, N/2+1, \cdots, N-1\}$。 证毕

定理1 将L-REP节点对应的编码码字$\boldsymbol{x}_v$的每两个元素分为一组，组成子向量组$\{\boldsymbol{x'}_{\mathrm{index}} : \mathrm{index} = 0, 1, \cdots, (N_v-2)/2\}$；若发送的信源比特$u_{\mathrm{mid0}} = 0$，子向量组$\boldsymbol{x'}_{\mathrm{index}}$中的元素相同；若发送的信源比特$u_{\mathrm{mid0}} = 1$，子向量组$\boldsymbol{x'}_{\mathrm{index}}$中的元素互异。

证明 由于$\boldsymbol{F}^{\otimes n} = \left[\boldsymbol{F}^{\otimes(n-1)}\ 0;\ \boldsymbol{F}^{\otimes(n-1)}\ \boldsymbol{F}^{\otimes(n-1)}\right]$，可将$\boldsymbol{f}_i (i \neq 0)$分为两部分：$\boldsymbol{f}_i = \left(\boldsymbol{f}_i^1, \boldsymbol{f}_i^0\right)$。其中，$\boldsymbol{f}_i^1$表示长度为$S$、循环长度为$S/2$的循环序列，$S$为2的整数次幂；$\boldsymbol{f}_i^0$表示长度为$N_v - S$的零序列。同理，$\boldsymbol{g}_i = \left(\boldsymbol{g}_i^1, \boldsymbol{g}_i^0\right)$。式(3)可以写成

$$\boldsymbol{x}_v = \sum_{i=0}^{N_v-1} u_i \boldsymbol{g}_i \tag{16}$$

由于二进制比特u_i的取值只可能为0或1，对二进制信源序列编码可以看作对生成矩阵的各行向量线性相加再取模二和。已知L-REP节点对应的信源序列$\boldsymbol{u}_v$前$N_v/2-2$位都传输零比特，代入式(16)可得

$$\boldsymbol{x}_v = \sum_{i=N_v/2-1}^{N_v-1} u_i \boldsymbol{g}_i \tag{17}$$

因此有

$$\mathrm{d}(\boldsymbol{x}_v) = \min\{\mathrm{d}(\boldsymbol{g}_i)\} \tag{18}$$

其中，$i = \{N_v/2-1, N_v/2, \cdots, N_v-1\}$。根据引理1及式(18)，假设$u_{\mathrm{mid0}} = 0$，式(17)可以写作

$$\boldsymbol{x}_v = \sum_{i=N_v/2}^{N_v-1} u_i \boldsymbol{g}_i \tag{19}$$

由于$\mathrm{d}\left(\boldsymbol{g}_i\right) \geq 2$且$\mathrm{d}\left(\boldsymbol{g}_i\right)$是2的整数次幂，则$\mathrm{d}(\boldsymbol{x}_v) \geq 2$且$\log_2 \mathrm{d}(\boldsymbol{x}_v) \in \mathrm{N}^+$；又因$\boldsymbol{g}_i = \left(\boldsymbol{g}_i^1, \boldsymbol{g}_i^0\right)$，经过若干个$g_i$线性相加并取模二和后，$\boldsymbol{x}_v$也可表示成由一个循环序列和一个零序列组合成的向量$\boldsymbol{x}_v = \left(\boldsymbol{x}^1, \boldsymbol{x}^0\right)$。因此子向量$\boldsymbol{x'}_{\mathrm{index}}$或者为全0向量或者为全1向量，即$\boldsymbol{x'}_{\mathrm{index}}$中的元素相同。假设$u_{\mathrm{mid0}} = 1$，由于此时$\boldsymbol{g}_i = [\ 1\quad 0\quad \cdots\quad 1\quad 0\ ]$，则$\mathrm{d}(\boldsymbol{x}_v) = 1$；同理可知$\boldsymbol{x'}_{\mathrm{index}} = [\ 1\quad 0\ ]$或$\boldsymbol{x'}_{\mathrm{index}} = [\ 0\quad 1\ ]$，即$\boldsymbol{x'}_{\mathrm{index}}$中的元素互异。 证毕

根据以上分析，得到L-REP节点的预译码算法设计方案，如图2所示。根节点下的左子节点为L-REP节点，记为$\mathrm{N}^{L\text{-REP}}$并对其进行预译码处理。虚线框中的过程表示$\mathrm{N}^{L\text{-REP}}$的预译码，主要包括硬判决处理、向量元素两两异或处理以及比特映射处理。该过程只在$\mathrm{N}^{L\text{-REP}}$这一层进行，没有涉及到下一层子树的遍历。已知预估计的比特位于第mid0位，经过预译码输出得到估计比特$\hat{u}_{\mathrm{mid0}}$。首先对$\boldsymbol{\alpha}_v$硬判决，得到硬判决序列$\mathbf{HD} = \mathrm{h}(\boldsymbol{\alpha}_v)$。然后将$\mathbf{HD}$中的元素按式(20)两两异或得到异或后的二进制序列$\mathbf{HD}'$

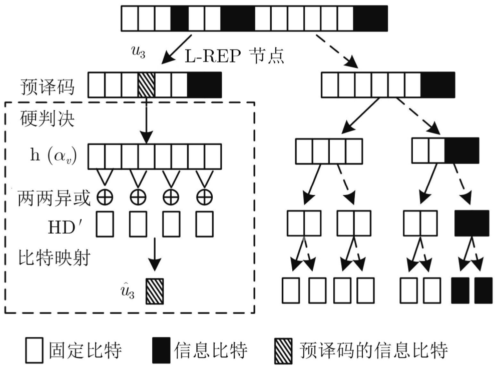


图 2 L-REP节点预译码设计

$$\mathbf{HD}'[i] = \mathbf{HD}[2i] \oplus \mathbf{HD}[2i+1] \tag{20}$$

其中，$i = \{0, 1, \cdots, N_v/2-1\}$。

为了得到$\mathbf{HD}$每两个元素相同的情况占整个向量的比例大小，将$\mathbf{HD}'$中所有元素相加，和记为count；然后进行比特映射处理：当$\mathrm{count} > N_v/2$时，$\hat{u}_{\mathrm{mid0}} = 1$；当$\mathrm{count} < N_v/2$时，$\hat{u}_{\mathrm{mid0}} = 0$。若$\mathrm{count} = N_v/2$，由于$\boldsymbol{\alpha}_v$是由LLR值组成，可以结合软判决译码的思想，用式(21)设计门限th。若$\mathrm{th} > 0$，则$\hat{u}_{\mathrm{mid0}} = 1$；反之，$\hat{u}_{\mathrm{mid0}} = 0$。

$$\begin{aligned}\mathrm{th} = \sum_{i=0}^{(N_v-2)/2} \{ & (1-2\mathbf{HD}'[i])(1-2\mathbf{HD}[2i])\boldsymbol{\alpha}_v[2i] \\ & +(1-2\mathbf{HD}'[i])(1-2\mathbf{HD}[2i+1])\boldsymbol{\alpha}_v[2i+1]\}\end{aligned} \tag{21}$$

3.2 L-BiREP节点的预译码处理

若节点v满足前mid1-1个叶子节点都为固定节点，第mid0，mid1个叶子节点为信息节点，则称节点v为L-BiREP节点。本小节将讨论对L-BiREP节点的第mid0位和第mid1位比特的预估计。

类比3.1节中的L-REP节点，L-BiREP节点对应生成矩阵的第mid1个行向量$\boldsymbol{g}_{\mathrm{mid1}}$可以由两个长度为$N_v/2$的向量表示：$\boldsymbol{g}_{\mathrm{mid1}} = \left(\boldsymbol{g}_{\mathrm{mid1}}^1, \boldsymbol{g}_{\mathrm{mid1}}^0\right)$，其中$\boldsymbol{g}_{\mathrm{mid1}}^1 = [\ 1\quad 0\quad \cdots\quad 1\quad 0\ ]$，$\boldsymbol{g}_{\mathrm{mid1}}^0$为全零向量。而对应生成矩阵的第mid0个行向量$\boldsymbol{g}_{\mathrm{mid0}} = [\ 1\quad 0\quad \cdots\quad 1\quad 0\ ]$。经过分析，$\boldsymbol{g}_{\mathrm{mid1}}$和$\boldsymbol{g}_{\mathrm{mid0}}$将影响到$\boldsymbol{x}_v$中“0”和“1”的排列结构。将$\boldsymbol{x}_v$分成两部分，表示成$\boldsymbol{x}_v = \left[\ \boldsymbol{x}_v^0\quad \boldsymbol{x}_v^1\ \right]$，分别将$\boldsymbol{x}_v^0$和$\boldsymbol{x}_v^1$中的元素每2个分为1组，记作向量$\boldsymbol{x}_{vi}^0$和$\boldsymbol{x}_{vi}^1 (0 \leq i < N_v/4)$。当$u_{\mathrm{mid1}}$和$u_{\mathrm{mid0}}$取不同值时，$\boldsymbol{x}_{vi}^0$和$\boldsymbol{x}_{vi}^1$遵循的规律如表1。

表 1　u_{mid1}和u_{mid0}对应$\boldsymbol{x}_{vi}^0$和$\boldsymbol{x}_{vi}^1$的取值情况

u_{mid1}	u_{mid0}	$\boldsymbol{x}_{vi}^0$	$\boldsymbol{x}_{vi}^1$
0	0	[1　1]/[0　0]	[1　1]/[0　0]
0	1	[0　1]/[1　0]	[0　1]/[1　0]
1	0	[0　1]/[1　0]	[1　1]/[0　0]
1	1	[1　1]/[0　0]	[0　1]/[1　0]

对于L-BiREP节点，已知输入的LLR信息向量$\boldsymbol{\alpha}_v$，经过预译码输出得到估计比特$\hat{u}_{\text{mid1}}$和$\hat{u}_{\text{mid0}}$。首先与L-REP节点进行同样的操作得到异或后的二进制序列$\mathbf{HD}'$。然后将$\mathbf{HD}$看作是由左右两个子向量组成，$\mathbf{HD}=(\mathbf{HD}_l,\mathbf{HD}_r)$；同样将$\mathbf{HD}'$看作由两个子向量组成，$\mathbf{HD}'=({\mathbf{HD}'}_l,{\mathbf{HD}'}_r)$；分别将${\mathbf{HD}'}_l$和${\mathbf{HD}'}_r$中的元素相加，记为$\text{count}_l$和$\text{count}_r$。再将$\text{count}_l$，$\text{count}_r$各自比较与$N_v/4$的大小：当$\text{count}_l > N_v/4$，$\text{count}_r > N_v/4$时，$\hat{u}_{\text{mid1}}=0$，$\hat{u}_{\text{mid0}}=1$；当$\text{count}_l < N_v/4$，$\text{count}_r < N_v/4$时，$\hat{u}_{\text{mid1}}=0$, $\hat{u}_{\text{mid0}}=0$；当$\text{count}_l < N_v/4$, $\text{count}_r > N_v/4$时，$\hat{u}_{\text{mid1}}=1$, $\hat{u}_{\text{mid0}}=1$；当$_l > N_v/4, \text{count}_r < N_v/4$时，$\hat{u}_{\text{mid1}}=1$，$\hat{u}_{\text{mid0}}=0$。出现相等的情况时，采用与3.2节一样的方法计算判决门限th_l和th_r，如式(22)和式(23)。当判决门限大于0时，说明对应的$\boldsymbol{x}_v^0$或$\boldsymbol{x}_v^1$中元素两两相异的概率更大；反之，$\boldsymbol{x}_v^0$或$\boldsymbol{x}_v^1$中元素两两相同的概率更大，然后根据表1确定$\hat{u}_{\text{mid1}}$和$\hat{u}_{\text{mid0}}$，得到L-BiREP的预译码结果。

$$\begin{aligned}\text{th}_l = &\sum_{i=0}^{(N_v-4)/4}\left\{(1-2{\mathbf{HD}'}_l[i])(1-2\mathbf{HD}_l[2i])\,\alpha_v[2i]\right.\\ &\left.+(1-2{\mathbf{HD}'}_l[i])(1-2\mathbf{HD}_l[2i+1])\,\boldsymbol{\alpha}_v[2i+1]\right\}\end{aligned} \tag{22}$$

$$\begin{aligned}\text{th}_r = &\sum_{i=0}^{(N_v-4)/4}\left\{(1-2{\mathbf{HD}'}_r[i])(1-2\mathbf{HD}_r[2i])\right.\\ &\cdot\boldsymbol{\alpha}_v[N_v/2+2i]\\ &+(1-2{\mathbf{HD}'}_r[i])(1-2\mathbf{HD}_r[2i+1])\\ &\left.\cdot\boldsymbol{\alpha}_v[N_v/2+2i+1]\right\}\end{aligned} \tag{23}$$

3.3 PDM-SSC算法设计

预译码处理只得到了组合码部分信息位的传输比特，因此本小节参考Chase译码算法[16]中的候选码枚举的方法，选择具有最大似然值的码字作为译码输出，完成组合码的完整译码。首先将预译码处理后的比特位看作已知位，根据组合码的结构构造候选码组$\mathcal{C}'=\left\{\boldsymbol{c}_i: 0\le i<2^{k_v}\right\}$，$k_v$表示组合码中右子节点信息比特的个数。每一个候选码字$\boldsymbol{c}_i$的LLR可表示为

$$l(\boldsymbol{c}_i)=\sum_k(1-2c_i[k])\,\alpha_v[k] \tag{24}$$

最大似然译码输出可表示为

$$\hat{\boldsymbol{u}}=\underset{\boldsymbol{c}_i\in\mathcal{C}'}{\operatorname{argmax}}\, l(\boldsymbol{c}_i) \tag{25}$$

PDM-SSC译码算法过程如图3所示。

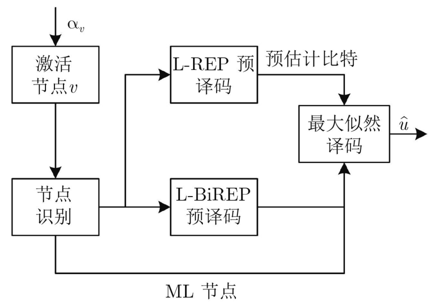


图 3 PDM-SSC译码框架

步骤1　判断节点v是属于L-REP节点、L-BiREP节点还是其它类型的节点；

步骤2　若v为L-REP节点，则将该节点和对应的$\boldsymbol{\alpha}_v$输入L-REP预译码模块，获得预译码比特$\hat{u}_{\text{mid0}}$；若v为L-BiREP节点，则将该节点和对应的$\boldsymbol{\alpha}_v$输入L-BiREP预译码模块，获得预译码比特$\hat{u}_{\text{mid0}}$和$\hat{u}_{\text{mid1}}$；若以上情况都不符合，则直接进行最大似然译码；

步骤3　将步骤2中预估计比特输入到最大似然译码模块中，根据剩余信息位构造候选码组$\mathcal{C}'$，再通过式(24)和式(25)选择具有最大似然值的码字作为最终的译码输出。

4　实验仿真及分析

下面将从理论分析以及实验仿真两个方面，分别比较SSC, ML-SSC以及PDM-SSC算法在AWGN信道中的译码时延及误码性能。

时延分析：本文参考文献[9]采用时间周期衡量算法的译码时延。文献[9]假定在译码过程中具有足够多的计算单元，可以在一个时间周期内完成软信息向量α_v的计算，而β_v以及硬判决操作看作立即完成。本文假定只有P个计算单元，P是2的整数次幂。每当一个节点被激活，计算其α_v至少需要消耗1个时间周期。由于1个计算单元只能处理1个实数的加减运算或大小比较操作，若计算α_v时需要的计算单元超过P，则按照超出的个数增加时间周期。其它相关的比特运算及硬判决运算同样看作不带来时延。最终，总的时间周期代表译码的时延大小。

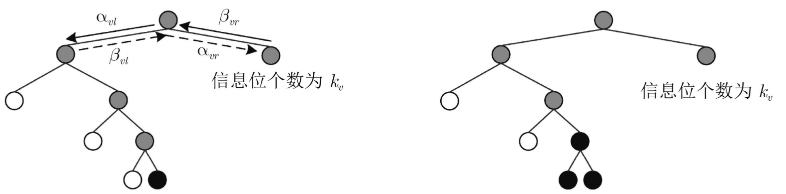
对于长度为N_v，信息位个数为k_v+1的L-REP节点来说，其左子节点实际上是一个REP节点，右子节点的类型未知，如图4(a)。在传统的SSC算法

中，当L-REP节点被激活，要完全译码左子节点需要遍历其左子树至叶子节点。首先至少需要1个时间周期计算出左子节点的软信息向量α_{vl}。根据REP节点结构特性，左子树下每层都有1个rate-R节点直到最后1个信息叶子节点，遍历到该信息叶子节点时需要计算$\log_2 N_v$个软信息向量。因此得到L-REP节点的$\boldsymbol{\beta}_{vl}$向量至少需要消耗$\log_2 N_v$个时间周期。然后计算L-REP节点的$\boldsymbol{\alpha}_{vr}$需要消耗1个时间周期，假设得到$\boldsymbol{\alpha}_{vr}$后再等待τ_r个时间周期直到右子树返回$\boldsymbol{\beta}_{vr}$，译码共需要消耗的时间周期至少为$\log_2 N_v + \tau_r + 2$。根据文献[15]，ML-SSC对L-REP节点的译码涉及到$\left(2^{k_v+1}+1\right)(N_v-1)$个运算，当计算单元个数为$P$时，需要的时间周期为$\left(2^{k_v+1}+1\right)(N_v-1)/P$个。本文提出的PDM-SSC算法，通过预译码可以先得到左子树的译码比特，预译码过程中的硬判决、对硬判决向量中的二进制元素进行异或操作看作立即完成。确定预译码比特需要经过比特映射，从实验上看，比特映射可以得到绝大多数的预译码比特的值，只有较小概率需要进行式(21)—式(24)的阈值判定得到预译码结果。因此将这些处理过程看作1个运算。预译码后，进行的最大似然译码的运算个数为$\left(2^{k_v}+1\right)(N_v-1)$个，最终PDM-SSC译码算法的平均时间周期为$\left(\left(2^{k_v}+1\right)\cdot(N_v-1)+1\right)/P$个。

对于长度为N_v，信息位个数为k_v+2的L-BiREP节点，其左子节点的最后两位传输信息比特，右子节点的类型同样未知，如图4(b)。假若用SSC算法译码，与L-REP节点译码的不同之处是遍历到左子树的倒数第2层的信息节点时就可立即返回码字$\boldsymbol{\beta}$，得到L-BiREP节点的$\boldsymbol{\beta}_{vl}$向量至少消耗$\log_2 N_v - 1$个时间周期，译码至少需要消耗$\log_2 N_v + \tau_r + 1$个时间周期。ML-SSC算法对L-BiREP节点的译码与对L-REP节点一样，只是L-BiREP节点的信息比特个数为k_v+2个，因此需要$\left(2^{k_v+2}+1\right)(N_v-1)/P$个时间周期。本文提出的算法对L-BiREP节点的译码与L-REP节点类似，预译码后可得到左子节点下的两位信息位的译码比特，总共需要$\left(\left(2^{k_v}+1\right)(N_v-1)+1\right)/P$个时间周期。

在只有P个计算单元的限制条件中，L-REP节点和L-BiREP节点在以上3种译码算法下的时延情况如表2。由于τ_r的大小取决于右子节点的具体结构，SSC译码算法在这两种节点下的译码时延并不确定，但可知需要激活L-REP节点(或L-BiREP节点)下的每一个结构为rate-R的子节点，其最小的译码时延为$\log_2 N_v + \tau_r + 2$(或$\log_2 N_v + \tau_r + 1$)。与SSC译码算法相比，本文提出的PDM-SSC译码算法的时延确定，不需要激活L-REP节点(或L-BiREP节点)下的子节点，时延不受右子节点的结构影响。与ML-SSC算法相比，PDM-SSC算法在L-REP节点的译码时延降低了约50%，而在L-BiREP节点中的译码时延降低了约75%。在整个的极化码译码中，一般来说计算单元的个数是固定的，相同的P下，PDM-SSC算法较于ML-SSC算法可译码更多类型，长度及码率更大的节点，因此推断PDM-SSC算法可以降低整体的译码时延。

仿真分析： 本文的实验仿真采用BPSK调制信号源信号，传输信道采用AWGN信道，极化码的编码方式参照文献[16]。在发送端，编码后的二进制序列$\boldsymbol{x}=(x_0,x_1,\cdots,x_{N-1})$映射成传输序列$\boldsymbol{s}=(s_0,s_1,\cdots,s_{N-1})$，映射规则为$\boldsymbol{s}=1-2\boldsymbol{x}$。在接收端得到接收向量$\boldsymbol{y}=(y_0,y_1,\cdots,y_{N-1})$，其中$\boldsymbol{y}=\boldsymbol{s}+w$，$w$表示均值$\mu=0$，方差为$\sigma^2$的高斯随机变量。仿真中使用到了长度$N=2048, 32768$的极化码，使用到的码率$R$包括0.5, 0.9两种。另外，



(a) L-REP 节点　　(b) L-BiREP 节点

图 4 L-REP节点和L-BiREP节点结构

表 2 P个计算单元下L-REP节点和L-BiREP节点的译码时延

算法	L-REP节点的译码时延	L-BiREP节点的译码时延
SSC	$\geq \log_2 N_v + \tau_r + 2$	$\geq \log_2 N_v + \tau_r + 1$
ML-SSC	$\left(2^{k_v+1}+1\right)(N_v-1)/P$	$\left(2^{k_v+2}+1\right)(N_v-1)/P$
PDM-SSC	$\left(\left(2^{k_v}+1\right)(N_v-1)+1\right)/P$	$\left(\left(2^{k_v}+1\right)(N_v-1)+1\right)/P$

实验中的时延都采用译码消耗的时间周期衡量。如无其它说明，计算单元个数$P=256$，信道噪声方差$\sigma^2=0.25$，PDM-SSC算法的译码节点信息比特的个数$k_v'\le 6$。

为了调查PDM-SSC算法在不同码长下的译码时延情况，本文采用2048和32768两种码长，码率均为0.5的两种极化码，分别用SSC以及PDM-SSC算法对其进行译码。译码时延情况如表3所示。假定时延增益表示PDM-SSC算法相对于其他算法译码时延降低的百分比。在相同的计算单元以及码率的条件下，与传统的SSC算法相比，对于较短或较长码长的极化码，PDM-SSC算法都可以有效地降低译码时延。当极化码码长为2048时，可降低25.8%的译码时延。

表 3　P=256, R=0.5时，本文算法与SSC算法时延情况对比

算法	码长	时延(时间周期)	时延增益(%)
SSC	2048	817	–
	32768	6973	–
PDM-SSC	2048	606	25.8
	32768	5731	17.8

为了调查PDM-SSC算法在不同码率下的译码时延情况，第2个实验采用$N=32768$，R分别为0.5, 0.9的两种极化码在PDM-SSC与ML-SSC这两种算法下进行译码。其中，ML-SSC译码算法中$k_v\le 4$，$N_v=16$。如表4所示，无论是在码率较小或是码率较大的情况下，本文方法与ML-SSC算法相比译码时延增益都有较明显的提升。译码节点个数表示对应算法可找到的简化译码的节点个数，当$R=0.5$时，ML-SSC算法可简化的节点个数为69个，而PDM-SSC算法可简化的节点比它提升了将近1倍，这是因为在同一个P下，PDM-SSC算法还可以对长度更长，信息位更多的节点译码，因此减少了更多子树的遍历。当$R=0.9$时，本算法可以比ML-SSC算法提升至18.7%的时延增益。

图5描述了$N=2048, R=0.5$的极化码在不同信噪比的AWGN信道下用PDM-SSC算法译码的误码率和误帧率。为了比较，在相同条件下分别用SSC和ML-SSC算法进行译码，并且ML-SSC译码

表 4　P=256, N=32768时，本文算法与ML-SSC算法时延情况对比

算法	码率	译码节点个数	时延(时间周期)	时延增益(%)
ML-SSC	0.5	69	6679	–
	0.9	60	3470	–
PDM-SSC	0.5	148	5731	14.2
	0.9	114	2822	18.7

算法中的参数$k_v\le 4$，$N_v=16$。由图5可知，3种算法的误码率曲线和误帧率曲线都十分接近。因此与经典的SSC译码算法相比，本文提出的PDM-SSC算法在译码过程中对误码性能的影响可以忽略。

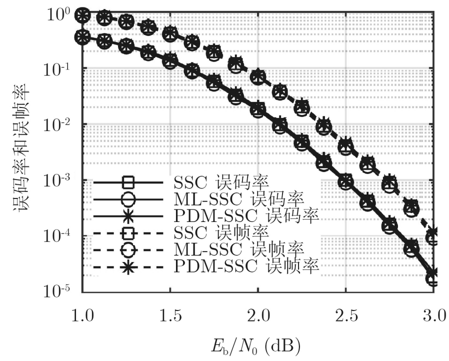


图 5 SSC, ML-SSC与PDM-SSC译码算法的误码率和误帧率

5　结束语

为了减少极化码译码过程中冗余节点的遍历，降低译码时延，本文提出了预译码的思想，在对整个节点译码之前，利用节点结构与中间似然值之间的关系先对中间的若干位比特译码，为接下来的最大似然译码减少计算量，从而可以在有限处理资源的条件下提高译码速率。该算法对译码节点结构的要求不高，不需要对每一种结构的节点设计不同的译码器，同时可以节约处理资源，比ML-SSC算法适用于更多的节点。由仿真结果可知，与SSC及ML-SSC算法相比，本文方法可在不影响误码性能的前提下有效降低极化码的译码时延。

参 考 文 献

[1] ARIKAN E. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels[J]. *IEEE Transactions on Information Theory*, 2009, 55(7): 3051–3073. doi: 10.1109/TIT.2009.2021379.

[2] İŞCAN O, BÖHNKE R, and XU Wen. Shaped polar codes for higher order modulation[J]. *IEEE Communications Letters*, 2018, 22(2): 252–255. doi: 10.1109/LCOMM.2017.2766621.

[3] 郭锐, 王美洁, 王杰. 基于缩短极化码的MLC NAND Flash差错控制技术研究[J]. 电子与信息学报, 2017, 39(7): 1658–1665. doi: 10.11999/JEIT160864.
GUO Rui, WANG Meijie, and WANG Jie. Research on the MLC NAND Flash error control technology based on polar codes[J]. *Journal of Electronics & Information Technology*, 2017, 39(7): 1658–1665. doi: 10.11999/JEIT160864.

[4] GULCU T C and BARG A. Achieving secrecy capacity of the wiretap channel and broadcast channel with a

confidential component[J]. *IEEE Transactions on Information Theory*, 2017, 63(2): 1311–1324. doi: 10.1109/TIT.2016.2631223.

[5] 朱鸿斌, 戴胜辰, 康凯, 等. 改进型极化码混合自动请求重传法[J]. 电子与信息学报, 2017, 39(5): 1136–1141. doi: 10.11999/JEIT160736.

ZHU Hongbin, DAI Shengchen, KANG Kai, *et al*. An improved HARQ scheme with polar codes[J]. *Journal of Electronics & Information Technology*, 2017, 39(5): 1136–1141. doi: 10.11999/JEIT160736.

[6] LEROUX C, RAYMOND A J, SARKIS G, *et al*. A semi-parallel successive-cancellation decoder for polar codes[J]. *IEEE Transactions on Signal Processing*, 2013, 61(2): 289–299. doi: 10.1109/TSP.2012.2223693.

[7] HUSMANN C, NIKOLAOU P C, and NIKITOPOULOS K. Reduced latency ML polar decoding via multiple sphere-decoding tree searches[J]. *IEEE Transactions on Vehicular Technology*, 2018, 67(2): 1835–1839. doi: 10.1109/TVT.2017.2761262.

[8] HASHEMI S A, CONDO C, and GROSS W J. A fast polar code list decoder architecture based on sphere decoding[J]. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2016, 63(12): 2368–2380. doi: 10.1109/TCSI.2016.2619324.

[9] ALAMDA-YAZDI A and KSCHISCHANG F R. A simplified successive-cancellation decoder for polar codes[J]. *IEEE Communications Letters*, 2011, 15(12): 1378–1380. doi: 10.1109/LCOMM.2011.101811.111480.

[10] SARKIS G, GIARD P, VARDY A, *et al*. Fast polar decoders: algorithm and implementation[J]. *IEEE Journal on Selected Areas in Communications*, 2014, 32(5): 946–957. doi: 10.1109/JSAC.2014.140514.

[11] HANIF M and ARDAKANI M. Fast successive-cancellation decoding of polar codes: identification and decoding of new nodes[J]. *IEEE Communications Letters*, 2017, 21(11): 2360–2363. doi: 10.1109/LCOMM.2017.2740305.

[12] HUANG Zhiliang, DIAO Chunjuan, DAI Jianxin, *et al*. An improvement of modified successive-cancellation decoder for polar codes[J]. *IEEE Communications Letters*, 2013, 17(12): 2360–2363. doi: 10.1109/LCOMM.2013.110413.132136.

[13] CHOI J and PARK I C. Improved successive-cancellation decoding of polar codes based on recursive syndrome decomposition[J]. *IEEE Communications Letters*, 2017, 21(11): 2344–2347. doi: 10.1109/LCOMM.2017.2730860.

[14] YOO H and PARK I C. Efficient pruning for successive-cancellation decoding of polar codes[J]. *IEEE Communications Letters*, 2016, 20(12): 2362–2365. doi: 10.1109/LCOMM.2016.2607167.

[15] SARKIS G and GROSS W J. Increasing the throughput of polar decoders[J]. *IEEE Communications Letters*, 2013, 17(4): 725–728. doi: 10.1109/LCOMM.2013.021213.121633.

[16] CHASE D. Class of algorithms for decoding block codes with channel measurement information[J]. *IEEE Transactions on Information Theory*, 1972, 18(1): 170–182. doi: 10.1109/TIT.1972.1054746.

刘建航：男，1978年生，副教授、博士，研究方向为信道编码，移动互联网.

何怡静：女，1994年生，硕士生，研究方向为信道编码.

李世宝：男，1978年生，副教授，研究方向为移动计算.

卢丽金：女，1992年生，硕士生，研究方向为信道编码.

邓云强：男，1993年生，硕士生，研究方向为信道编码.