

系统极化码低复杂度编码优化方案

马林华^{1,2}, 刘士平¹, 胡星¹, 黄天宇¹, 徐彬³

(1. 空军工程大学航空工程学院, 陕西 西安 710038;

2. 西安电子科技大学综合业务网国家重点实验室, 陕西 西安 710071;

3. 空军航空大学初级训练基地, 黑龙江 哈尔滨 150100)

摘 要: 为解决系统极化码在编码过程中因分步计算造成的时延和由循环迭代“异或”计算造成的计算复杂度, 提出并定义了降维裂解策略, 并由此提出了基于降维裂解策略的系统极化码并行编码算法, 然后在 AWGN 信道下进行了仿真验证和计算复杂度分析。结果表明, 与传统算法相比, 所提算法编码增益略优或基本保持一致, 但计算复杂度优化率最高可达 80.92%, 更适合于硬件实现与工程应用, 具有一定的实用价值。

关键词: 极化码; 系统极化码; 并行编码; 复杂度; 裂解; 误码率

中图分类号: TN911.22

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2018127

Optimizing low complexity encoding method for systematic polar code

MA Linhua^{1,2}, LIU Shiping¹, HU Xing¹, HUANG Tianyu¹, XU Bin³

1. Aeronautics and Astronautics Engineering College, Air Force Engineering University, Xi'an 710038, China

2. The State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China

3. Air Force Aviation University Primary Training Base, Harbin 150100, China

Abstract: In order to solve the delay caused by step-by-step calculation and the computational complexity caused by iterative “exclusive-or” computation during the encoding process, a dimensionality reduction strategy was proposed and defined. Based on this, system polarization code parallel coding algorithm for cracking strategy was proposed. Simulation and computational complexity analysis were carried out on AWGN channel. The results show that the coding gain of the above algorithm is slightly better than the traditional one or almost the same, but the computational complexity is up to 80.92%, which is more suitable for hardware implementation and engineering application. It is more suitable for hardware implementation and has a certain practical value.

Key words: polar code, systematic polar code, parallel encoding, complexity, splitting decomposition, bit error rate

1 引言

信道纠错编码技术是提高数字通信系统抗干扰能力的关键技术之一。香农在有噪信道编码理论中指出, 存在可以达到香农限的码字^[1]。自 2008 年土耳其教授 Arican 基于信道极化定理提出极化码

(polar code) 以来, 极化码凭借低复杂度编译码以及信道容量可达的优势在信道编码领域占据重要地位, 逐渐受到重视并开始应用于各个领域, 拥有良好的发展前景^[2-4]。

在编码理论中, 系统码是输出码字包含输入信息序列的一类编码, 即其信息比特会作为码字的一部分

收稿日期: 2018-01-11; 修回日期: 2018-05-19

基金项目: 国家自然科学基金资助项目 (No.61472442); 陕西省科技攻关基金资助项目 (No.2017GY-049); 航空科学基金资助项目 (No.20155896025)

Foundation Items: The National Natural Science Foundation of China (No.61472442), Shaanxi Province Scientific and Technological Project (No.2017GY-049), Aviation Science Foundation (No.20155896025)

直接显现出来，而非系统码的输出码字中不包含输入信息序列。极化码的标准形式是非系统码，鉴于任意线性码都可以转换成系统码，因此，极化码也可被系统地编码^[5]。研究表明，由于在译码时系统码的信息比特可以通过信道被直接观察，系统极化码相比于传统的非系统极化码具有更好的误比特率性能^[5]，且信息位可直接在系统码码字中表现出来，即不需要通过冗长的译码过程就能得到合理的准确估计，可快速确定所接收源符号的正确性^[6-7]。因此，系统极化码逐渐成为工程应用上的首选。

Arıcan^[5]在首次提出系统极化码的同时也给出了该系统极化码的串行编码算法，文献[8]针对此算法进行了补充和复杂度分析。该算法将码字拆分成2个部分并按序依次进行嵌套运算，由于该运算机制随着码长的增加，会提高2个计算步骤之间的错误传播概率，并在2个计算步骤之间因等待造成系统时延。针对此问题，文献[9-10]提出一种并行计算的编码算法，利用双极化结构取代串行编码算法中分步计算的过程，可以实现任意码长、码率下对编码码字的一步直接计算。虽然此算法可以实现高度并行计算，降低因分步计算等待造成的时延及错误传播，但是由于增加了一个额外的极化结构导致“异或”运算的次数随着码长增加而呈指数级增长，提高了系统的计算复杂度，降低了编码效率，因此并没有从根本上解决系统运算时延问题^[11]。基于上述问题，本文提出基于降维裂解策略的系统极化码并行编码算法，基于“分治法”的思想将整段码字裂解计算再合并，在保证可实时并行计算且不增加额外存储空间的同时，降低了计算复杂度及错误传播概率，更适用于硬件实现及实际应用。

2 系统极化码及背景算法描述

2.1 极化码与系统极化码

(N, K) 极化编码输入的 N bit信息序列 $u_1^N = (u_1, u_2, \dots, u_N)$ 由 K bit信息比特和 $(N - K)$ bit冻结比特(一般为全0)这2个部分组成。分别对 $N = 2^n$ 个独立的二进制离散无记忆信道(B-DMC) W 进行信道合成及信道分解的操作后,可得到 N 个独立等价子信道 $W_N^{(i)}, i = 1, 2, \dots, N$ 。根据信道极化定理,巴氏参数 $Z(W_N^{(i)}) \triangleq \sum_{y \in Y} \sqrt{W_N^{(i)}(y|0)W_N^{(i)}(y|1)}$ 表示通过该信道 $W_N^{(i)}$ 传输1 bit数据的错误概率上界, $Z(W_N^{(i)}) \in [0, 1]$ 。

信道容量 $I(W_N^{(i)})$ 与 $Z(W_N^{(i)})$ 满足:当且仅当 $Z(W_N^{(i)}) \approx 0$ 时, $I(W_N^{(i)}) \approx 1$;当且仅当 $Z(W_N^{(i)}) \approx 1$ 时, $I(W_N^{(i)}) \approx 0$ ^[2]。经过计算, $I(W_N^{(i)}) \approx 1$ 的信道视作无噪信道,用作信息位传输信息比特;而 $I(W_N^{(i)}) \approx 0$ 的信道视作全噪信道,用作冗余位传输固定的冻结比特。

根据上述信息位及冗余位的选取方法, u_1^N 中的 K bit信息比特通过信息位传输, $(N - K)$ bit冻结比特通过冗余位传输,依据式(1)完成极化编码,其中, G_N 为生成矩阵。

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

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

其中,信息位索引矩阵 A 为信息信道的下标集合; A^c 为其补集,即冻结信道的下标集合,故 u_A 和 u_{A^c} 分别为输入码字矩阵中信息比特及冻结比特子向量;符号“ \oplus ”为模二加运算;“ \otimes ”为张量积运算;根据极化结构, $B_N = R_N(I_2 \otimes B_{N/2})$ 为实现索引位逆转操作的矩阵; $F = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$ 为极化变换的核^[2];

R_N 为比特翻转矩阵,可实现元素的奇偶分离重排。由此可见,极化码也是线性分组码的一种,

(N, K) 极化码的码率 $R = \frac{K}{N}$ 。极化编码可视作根据信道可靠度的计算选取合适的信道,确定编码参数 (N, K, A, u_{A^c}) 的 G_N 陪集编码过程。

令信息序列 $u_1^N = (u_A, u_{A^c})$,码字 $x_1^N = (x_B, x_{B^c})$,其中, B 为 $\{1, 2, \dots, N\}$ 的任意子集,在这里设置 $A = B$ 。则非系统码编码计算式,即式(1)可转化为

$$x_B = u_A G_{AB} + u_{A^c} G_{A^c B} \quad (3)$$

$$x_{B^c} = u_A G_{AB^c} + u_{A^c} G_{A^c B^c} \quad (4)$$

其中, G_{AB} 是由生成矩阵 G_N 中元素 $G_{i,j}$ 组成的子矩阵, $i \in A, j \in B$,其余子矩阵构造方式与之类似。 x_B 等价于非系统编码中的 u_A 即输入码字矩阵的信息位子向量,系统极化编码信息位及冗余位的确定方式与上述非系统码一致。由系统码的性质可知,一般情况下 x_B 已知,则由式(5)可计算得到 u_A ,即

$$u_A = (x_B - u_{A^c} G_{A^c B})(G_{AB})^{-1} \quad (5)$$

得到 u_A 后代入式(4)即可得到 x_{B^c} 。

综上, 对任意非系统极化码编码器 (A, u_A) , 当 A 与 B 中元素数量相等且 G_{AB} 为可逆矩阵时, 存在与之对应的系统极化码编码器 (B, u_B) , 且系统编码的码字为 $x_1^N = (x_B, x_{B^c})$ 。算法可看作多路径同时进行的串行抵消 (SC) 译码算法。

2.2 系统极化码并行编码

文献[5]中串行系统极化码编码器在进行循环迭代编码运算时可用计算式表达为下述过程。给定信息比特向量 u_A , 通过对信息位的预处理得到即将输入编码器中的信息序列向量 $v = (v_1, v_2, \dots, v_n)$, 其中, $i \in A$ 时, $v_i = 0$ 。

根据极化结构自身的循环迭代性, 将序列 x 和 v 拆分成元素数量相同的 2 个部分表示, 即 $v = (v', v'')$, $x = (x', x'')$ 。编码过程可表示为

$$x'' = v'' F^{\otimes(n-1)} \quad (6)$$

$$x' = v' F^{\otimes(n-1)} + x'' \quad (7)$$

可以看出, 每一步的循环迭代只有通过式(6)计算出 x'' 后才可代入式(7)计算 x' , 从而得到码字 x 。因此, 随着码长的增加系统的时延也会增加, 该算法的效率也会随之受到严重影响, 同时在式(6)中出现的编码错误也会随之代入式(7)的计算中, 增加了错误传播的概率。

针对上述问题, 文献[9]对此进行改进并提出了一种新型系统编码器, 该编码器可以实现编码的并行计算, 并可直接经过编码得到最终码字, 避免中间步骤的等待时延。

定义矩阵 $E = (E_{i,j})_{k \times n}$, 其中, $E_{i,j} = \begin{cases} 1, & j = a_i \\ 0, & \text{其他} \end{cases}$,

矩阵 $A = \{a_j\}_{j=0}^{k-1}$, 其中, $1 \leq a_1 < a_2 < \dots < a_k \leq n$ 。信息位索引矩阵 A 中元素 a_j 为对应信息信道下标的索引。矩阵 E 的作用是将其左乘到其他矩阵后, 只保留信息位索引下的行。该并行编码算法可简述为以下 4 步。

步骤 1 将矩阵 E 右乘信息比特向量 u_A , 得到长度为 n 的向量 v_1 。该向量中, 由 a_j 索引的列向量等于 $(u_A)_j$, 其他列全部为 0, 即

$$v_1 = u_A E \quad (8)$$

步骤 2 将 v_1 进行第一次极化处理得到 v_2 , 即

$$v_2 = v_1 F^{\otimes n} \quad (9)$$

步骤 3 将 v_2 中非 a_j 索引的列向量全部置 0, 得到向量 v_3 , 即

$$v_3 = v_2 E^T E \quad (10)$$

步骤 4 将 v_2 进行第二次极化处理得到生成码字 x , 即

$$x = v_3 F^{\otimes n} \quad (11)$$

综合式(8)~式(11), 可以整合出生成码字计算式, 即

$$x = u E F^{\otimes n} E^T E F^{\otimes n} \quad (12)$$

该并行编码算法的编码器结构如图 1 所示。由式(12)及图 1 可以看出, 该编码算法将输入码字看作一个整体对编码过程完成“一步计算”, 解决了系统极化码串行编码因分段计算造成的时延及错误传播问题。

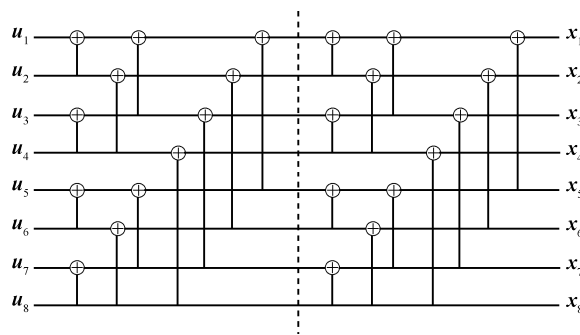


图 1 系统极化码并行编码算法的编码器结构

3 降维裂解并行编码算法

3.1 降维裂解策略

上述并行编码算法虽然解决了传统系统极化码循环迭代编码过程中存在的时延问题, 但是通过图 1 可以看出, 该编码算法相当于虚线左、右两侧 2 个极化结构的级联, 即在每一次编码计算的过程中都会比传统的编码算法多进行一次极化运算, 即与 $F^{\otimes n}$ 进行 2 次乘法运算。 $F^{\otimes n}$ 是下三角矩阵, 由高斯近似 (GE) 可知, 在循环递归编码过程中每解决一次 $F^{\otimes n}$ 运算需要进行“异或 (XOR)”运算的次数近似为 $3^{\text{lb}N} = N^{\text{lb}3} \approx N^{1.585}$ 次^[12-13]。当码长 N 逐渐增加时, 编码计算复杂度以及硬件存储复杂度会呈指数级增长, 远超文献[5]中提出的可实现的 XOR 复杂度 $\Theta(N \log N)$ 。因此, 该算法虽然解决了因等待而造成的编码时延及错误传播问题, 但是却增加了运算复杂度, 降低了编码效率。

在计算机科学中，“分治法”迭代策略的核心思想是将规模较大的问题分解成多个独立等属性的小规模问题并递归解决，从而达到降低计算复杂度与时延的目的。基于该思想，针对系统极化码并行编码算法，本文提出了降维裂解策略，核心要义是基于编码码树将码源信息序列合理拆分并递归计算，从而解决编码计算“ $F^{\otimes n}$ ”运算复杂度高的问题。

定义编码码字按区间的降维裂解策略。假设有一组编码码字向量 $\mathbf{x}=[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ ，基于编码码树，在码树第 l 层将码字裂解拆分成 2^l 个长度为 2^{n-l} 的等长度码字片段，即

$$\mathbf{x}=[\mathbf{x}_{2^l}^l, \dots, \mathbf{x}_2^l, \mathbf{x}_1^l], \quad 1 \leq l \leq n \quad (13)$$

其中，元素 $\mathbf{x}_i^l=[\mathbf{x}_{\alpha_i}, \dots, \mathbf{x}_{\alpha_i+2^{n-l}-1}]$ 表示码树第 l 层第 i 个子向量， $\alpha_i=(2^l-i)2^{n-l}+1, 1 \leq i \leq 2^l$ 。 \mathbf{x}_i^l 可理解为类似于 C 语言中码字向量 \mathbf{x} 子向量的指针。定义偏移向量 $\{p_{l,i}: 0 \leq l \leq n, 1 \leq i \leq 2^l\}$ ， l 与 i 分别表示图 2 所示循环递归编码码树的层数及各层中向量的序号。根据裂解策略及编码码树观察可得出，第 l 层的 2^l 个向量长度均为 2^{n-l} ，且每层的向量按顺序连接起来均为长度相同的 N 码字向量，这也验证了上述定义的裂解算法的正确性。

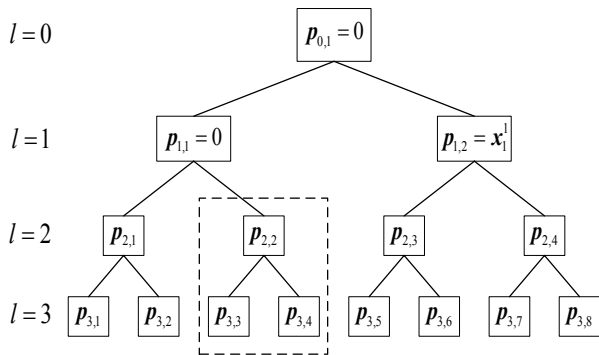


图 2 循环递归编码码树结构

假设 \mathbf{B} 为一组由码源信息经过一系列变换得到的向量， \mathbf{x} 为 \mathbf{B} 经过极化运算结构后得到的全新编码码字，即

$$\mathbf{x}=\mathbf{B}\mathbf{F}^{\otimes n} \quad (14)$$

式(14)为系统码编码算法的核心计算式，下面分别就 $l=1$ 和 $2 \leq l \leq n$ 这 2 种情况详细阐述基于上述降维裂解策略的低复杂度优化编码算法的工作流程。

情况 1 当循环至 $l=1$ 时，根据上述裂解策略，

令 $\mathbf{x}=[\mathbf{x}_2^1, \mathbf{x}_1^1]$ ， $\mathbf{B}=[\mathbf{B}_2^1, \mathbf{B}_1^1]$ ，式(13)可裂解为

$$\mathbf{x}_1^1=\mathbf{B}_1^1\mathbf{F}^{\otimes n-1} \quad (15)$$

$$\mathbf{x}_2^1=\mathbf{B}_2^1\mathbf{F}^{\otimes n-1}+\mathbf{x}_1^1 \quad (16)$$

对比式(15)与式(16)，后者在计算 \mathbf{x}_2^1 时加上了偏移向量 \mathbf{x}_1^1 ，因此，定义第一层的偏移向量，即初始值 $p_{1,1}=\mathbf{0}$ ， $p_{1,2}=\mathbf{x}_1^1$ 。

情况 2 当循环至 $2 \leq l \leq n$ 时，根据所提算法，在 $l-1$ 层会得到向量 \mathbf{x} 与 \mathbf{B} 按裂解策略划分得到的不同区间相互关联的 2^{l-1} 个等式，即

$$\mathbf{x}_i^{l-1}=\mathbf{B}_i^{l-1}\mathbf{F}^{\otimes n-l+1}+p_{l-1,i}, \quad 1 \leq i \leq 2^{l-1} \quad (17)$$

则在 l 层，上述每一个等式类比式(15)与式(16)裂解为

$$\mathbf{x}_{2i-1}^{l-1}=\mathbf{B}_{2i-1}^{l-1}\mathbf{F}^{\otimes n-l}+p_{l,2i-1} \quad (18)$$

$$\mathbf{x}_{2i}^{l-1}=\mathbf{B}_{2i}^{l-1}\mathbf{F}^{\otimes n-l}+p_{l,2i} \quad (19)$$

偏移向量 $p_{l-1,i}$ 按式(13)所定义的裂解策略二等分成 $(p_{l-1,i})_1^l$ 和 $(p_{l-1,i})_2^l$ ，则 l 层的偏移向量 $\{p_{l,2i-1}, p_{l,2i}\}$ 更新计算式为

$$p_{l,2i-1}=(p_{l-1,i})_1^l \quad (20)$$

$$p_{l,2i}=(p_{l-1,i})_1^l+(p_{l-1,i})_2^l+\mathbf{x}_{2i-1}^l \quad (21)$$

式(15)~式(21)就是基于降维裂解策略对编码过程式(14)进行优化的全过程，其主要目的是通过不断的裂解、降维，使 $F^{\otimes n}$ 也随之降维成 $\begin{pmatrix} F^{\otimes n-1} & 0 \\ F^{\otimes n-1} & F^{\otimes n-1} \end{pmatrix}$ 并拆分运算。整体上将一整段码字拆分成多个片段运算，从而达到减少计算量的目的。下面基于编码码树具体阐述编码的循环迭代步骤。

图 2 虚线框内代表上述降维裂解编码算法的一个基本循环计算单元，每个基本单元的具体循环迭代步骤如图 3 所示。假设有一组 $n+1$ 层完整的编码码树，层数 $l=0, \dots, n$ 。第 $l-1$ 层有 2^{l-1} 个节点并分别关联着由式(17)表示的 2^{l-1} 组 $2^{n-(l-1)}$ 个等式。由图 3 可知， $l-1$ 层中每个节点都在 l 层分裂成 2 个节点。最终，式(14)通过整个编码码树的按序遍历得到最终结果即编码码字。每次遍历至叶子节点（即 $l=n$ ）时，得到等式 $\mathbf{x}_i^n=\mathbf{y}_i^n+p_{n,i}$ ，其中， \mathbf{x}_i^n 或 \mathbf{y}_i^n 未知，偏移向量 $p_{n,i}$ 根据式(20)和式(21)进行实时更新。每组码树中除叶子节点只历经一次外，其余子节点都会历经 3 次。

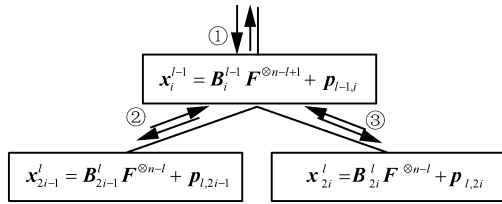


图 3 循环迭代单元结构

第一次 (①): 所需的偏移向量 $p_{l,2i-1}$ 由式(20)进行复制计算得到, 然后循环至左侧分支, 根据式(18)进行计算得出结果。

第二次 (②): 由该节点左侧分支的计算, 可以得到 x_{2i-1}^l , 因此, 可通过式(21)对 $p_{l,2i}$ 进行实时更新。然后循环至右侧分支, 根据式(19)进行计算得出结果。

第三次 (③): 第三次也是最后一次返回该节点, 此时已经遍历了整个子树并得到了式(17)的结果, 最终返回父节点。

在整个编码码树遍历结束后返回根节点, 并得到循环迭代后系统极化码的编码码字, 同时也完成了整个编码过程。可以看出, 该降维裂解策略的核心是利用“分治法”的思想对式(14)进行处理, 将码字长度较长、计算复杂度以及存储复杂度较高的问题利用不断分解的方式弱化, 即不断弱化 $F^{\otimes n}$ 造成的影响, 将裂解后的片段分别进行循环迭代运算从而组合成完整的编码码字。

3.2 基于降维裂解策略的系统极化码并行编码

利用上述定义的降维裂解编码策略, 在保留文献[8]中并行编码算法优点的基础上, 对式(12)进一步做低复杂度优化, 提出了系统极化码的降维裂解并行编码, 具体算法描述如下。

并行算法编码过程中共存在 2 个极化过程, 在编码式(12)中体现为与 2 个“ $F^{\otimes n}$ ”相乘。因此, 利用上述定义的裂解降维策略依次对 2 个极化结构(即 $F^{\otimes n}$) 进行低复杂度优化, 得到基于降维裂解策略的并行编码计算式为

$$x = \underbrace{uE}_{B'} F^{\otimes n} \underbrace{E^T E}_{B''} F^{\otimes n} \quad (22)$$

式(22)所表示的优化并行编码算法可具体阐述如下。

步骤 1 令式(14)中的 $B = B' = uE$, 利用上述降维裂解算法计算后得到

$$x' = B' F^{\otimes m} = (uE) F^{\otimes m} \quad (23)$$

步骤 2 将步骤 1 中得到的序列 x' 按照传统并

行编码算法中间步骤进行整合, 将非信息位全部置 0 得到 $B'' = x' E^T E$ 。继续利用降维裂解算法计算得到最终的编码码字, 即

$$x = B'' F^{\otimes m} = (x' E^T E) F^{\otimes m} \quad (24)$$

上述 2 个步骤(即式(23)和式(24))是依次进行的, 利用所提降维裂解策略对传统并行编码算法进行低复杂度优化, 充分保留了传统并行算法“一步计算”降低错误传播的特点, 最终得到基于降维裂解策略的并行编码。

4 仿真验证及复杂度分析

在 AWGN 信道下分别选取不同码长的随机生成码字, 并利用上述降维裂解并行编码算法进行编码。经过二进制频移键控(BPSK)调制后, 在接收端均采用 SC 译码算法进行信道译码^[14-15]。由于系统编码的性质, 要求编码后对应信息位均为原始给定的信息, 因此, 为验证本文编码算法的可靠性, 在判定误码率时采用的方式与非系统极化码误码率计算不同, 而是采用将接收端接收到的码字 y 的信息位索引信息 y_A 直接与原始给定信息 u_A 进行对比的方式计算误码率。图 4 与图 5 分别表示码长为 256 和 1 024、码率为 $\frac{1}{2}$ 的码字在不同编码方式下的误码率(BER,

bit error rate) 曲线。对比分析可以看出, 正如文献[5]中所讲, 系统极化码在译码时信息比特可以通过信道被直接观察到, 因此, 相比于非系统编码具有更好的抗误码性能, 且随着码长的增加效果更加显著。并行编码算法是由输入信息码字到生成码字的一步直接计算, 解决了传统串行编码因等待分步计算编码过程

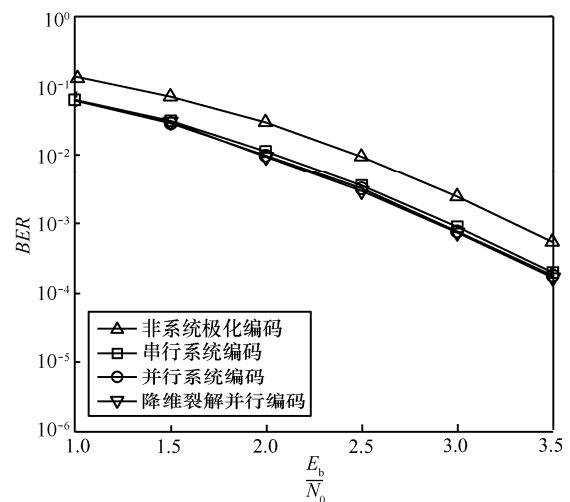


图 4 (256,128) 系统极化码不同编码算法性能对比

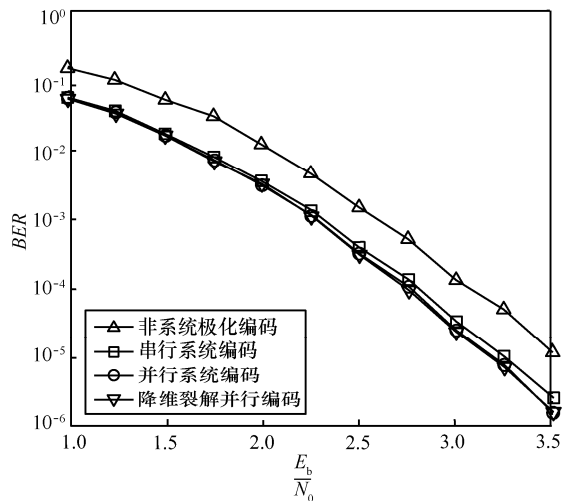


图 5 (1024,512) 系统极化码不同编码算法性能对比

中嵌套计算产生的错误传播问题，使误码率稍有改进并随码长和信噪比的提升性能更优越。由于上文所述基于降维裂解的并行编码算法保留了并行编码算法的这一优势，因此，可以看出，该算法的误码率与并行编码相比极为接近，但是相比之下却在计算复杂度方面做出了很大改善。

定义函数 $f(N)$ 表示码长为 N 时所需要进行 XOR 计算的次数，上述编码算法本质上是一种由 2 个降维裂解极化结构级联组成的循环迭代编码方式，因此根据式(18)~式(21)可得到降维裂解并行编码算法 XOR 计算次数的递归计算式为

$$f(N) = 2 \left[N + 2f\left(\frac{N}{2}\right) \right], \quad N = 2^n \text{ 且 } f(1) = 2 \quad (25)$$

式(25)等号右侧括号内第一个加数“ N ”是赋予偏移变量循环迭代即式(20)和式(21)计算的存储空间， $f\left(\frac{N}{2}\right)$ 是根据编码过程中的裂解迭代得到的，从而得到 $f(N)$ 与 $f\left(\frac{N}{2}\right)$ 之间的关系。可计算得到递归式(25)的闭合表达式，即利用上述降维裂解并行编码算法对码长等于 N 的码字循环迭代，编码所需 XOR 计算总数为

$$f(N) = 2N(1 + \lg N) \quad (26)$$

对比其他传统编码算法所需 XOR 计算次数，可得到编码复杂度对比如表 1 所示，不同码长下本文算法与传统编码算法计算复杂度优化率对比如表 2 所示，不同编码算法复杂度变化趋势如图 6 所示。由图 2 所示编码码树结构可以发现，从根节点至叶子节点的遍历过程中，任一节点调用循环迭代函数

次数最多为 $n+1$ 次。因此，该编码系统分配存储空间至多为 $N + \frac{N}{2} + \dots + 1 = (2N - 1)$ bit。而其余的编码

算法不需要不断调用函数循环迭代，因此，只需赋予编码计算中间 2 个步骤存储位置，即 $2N$ bit。由表 2 和图 6 可知，随着码长的增加，本文降维裂解算法与串行和并行系统编码算法相比，计算复杂度明显降低，优化率逐渐提高，当 $N=1024$ 时，在保证编码可靠性（即误码率基本一致）的前提下，优化率分别提高至 61.85%和 80.92%。但是从图 6 可以发现，当码长 $N < 128$ 时，由于降维裂解并行算法特有的循环迭代特性，其计算复杂度比传统串行编码略高，但是由图 4 和图 5 可以看出，由于本文算法保留了并行算法低错误传播的优点，虽然复杂度略高，但编码增益却优于串行编码。

表 1 系统极化码编码复杂度对比

编码算法	占用存储空间/bit	XOR 计算次数
非系统化编码	$2N$	$\frac{N}{2} \lg N$
串行系统编码	$2N$	$N^{1.585}$
并行系统编码	$2N$	$2N^{1.585}$
降维裂解编码	$2N - 1$	$2N(1 + \lg N)$

表 2 不同码长下本文算法与传统编码算法计算复杂度优化率对比

码长 N	与串行系统编码相比优化率	与并行系统编码相比优化率
128	6.37%	53.19%
256	29.7%	64.89%
1024	61.85%	80.92%

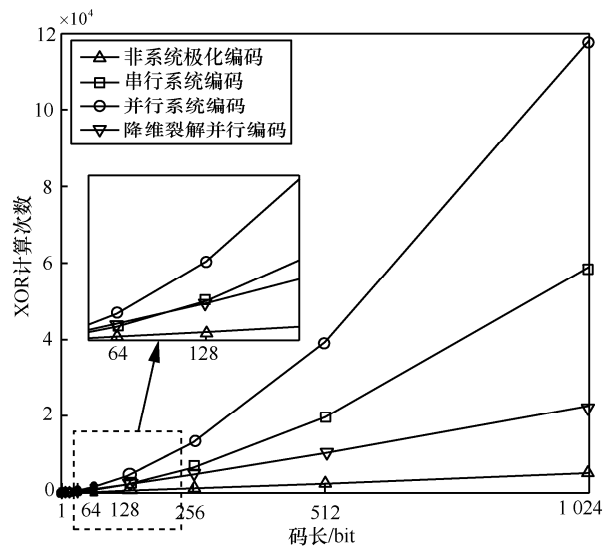


图 6 不同编码算法复杂度变化趋势

上述仿真验证及复杂度分析表明, 当码长 $N > 128$ 时, 降维裂解算法的编码增益与并行系统编码基本一致, 但是计算复杂度明显降低。相比于串行编码, 无论是计算复杂度还是编码增益, 都起到了优化的效果。当码长 $N < 128$ 时, 编码增益与上述类似, 但计算复杂度略高于串行系统编码。由于码长较短使计算复杂度提升并不明显, 且编码增益优于串行编码, 因此本文算法在任意码长下都可以发挥作用。

5 结束语

本文回顾了极化码传统编码与系统编码原理以及系统编码的优点, 简要介绍了现有系统极化码串行和并行编码算法原理并分别分析论证了各自的优缺点。在充分利用并行编码算法相比于串行编码高效率、低时延及低错误传播概率的优点, 为降低系统计算复杂度, 提出了基于降维裂解策略的并行编码算法, 并推导出该算法的循环迭代编码递归式。最后通过仿真验证了所提编码算法的合理性, 并与现有编码算法进行了复杂度对比分析。结果表明, 当码长 $N \geq 128$ 时, 本文所提算法与并行编码算法的编码增益基本一致且优于串行编码算法, 计算复杂度明显降低。随着信噪比的增加, 本文所提算法与串行系统编码相比, 编码增益更为突出。因此, 本文算法降低了“异或”计算复杂度及错误传播概率, 保证了编码的可靠性, 更适用于实际系统应用与硬件实现。

参考文献:

- [1] SHANNON C E. A mathematical theory of communication[J]. Bell System Technical Journal, 1948, 19(4): 271-285.
- [2] ARICAN 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.
- [3] ARICAN E. A performance comparison of polar codes and reed-muller codes[J]. IEEE Communications Letters, 2008, 12(6): 447-449.
- [4] ARICAN E. Channel combining and splitting for cutoff rate improvement[J]. IEEE Transactions on Information Theory, 2006, 52(2): 628-639.
- [5] ARICAN E. Systematic polar coding[J]. IEEE Communications Letters, 2011, 8(15): 860-862.
- [6] FENG B, ZHANG Q, JIAO J. An efficient rateless scheme based on the extendibility of systematic polar codes[J]. IEEE Access, 2017, PP(99): 1.
- [7] YOO H, PARK I C. Partially parallel encoder architecture for long polar codes[J]. IEEE Transactions on Circuits & Systems II Express Briefs, 2015, 62(3): 306-310.
- [8] LI L, ZHANG W. On the encoding complexity of systematic polar codes[C]//IEEE International System-on-Chip Conference. 2015: 415-420.
- [9] SARKIS G, TAL I, GIARD P, et al. Flexible and low-complexity encoding and decoding of systematic polar codes[J]. IEEE Transactions on Communications, 2016, 7(65): 2732-2745.
- [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.
- [11] TAL I, VARDY A. How to construct polar codes[J]. IEEE Transactions on Information Theory, 2011, 59(10): 6562-6582.
- [12] RICHARDSON T J, SHOKROLLAHI M A, URBANKE R. Design of capacity-approaching irregular low-density parity-check codes [J]. IEEE Transaction on Information Theory, 2001, 47(2): 619-637.
- [13] CHUNG S Y, RICHARDISON T J, URBANKE R. Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation[J]. IEEE Transactions on Information Theory, 2001, 47(2): 657-670.
- [14] ZHANG Z Y, ZHANG L, WANG X B, et al. A split-reduced successive cancellation list decoder for polar codes[J]. IEEE Journal on Selected Areas in communications, 2016, 34(2): 292-302.
- [15] BALATSOUKAS-STIMMING A, RAYMOND A J, GROSS W J, et al. Hardware architecture for list successive cancellation decoding of polar codes[J]. IEEE Transactions on Circuits & Systems II Express Briefs, 2014, 61(8): 609-613.

[作者简介]



马林华 (1965–), 男, 陕西汉中, 博士, 空军工程大学教授、博士生导师, 主要研究方向为抗干扰通信、信道编码、无线自组织网络。



刘士平 (1994–), 男, 黑龙江哈尔滨人, 空军工程大学硕士生, 主要研究方向为信道编码、极化码、抗干扰通信。

胡星 (1990–), 男, 河南南阳人, 空军工程大学博士生, 主要研究方向为模拟量编码、卫星通信。

黄天宇 (1993–), 男, 辽宁营口人, 空军工程大学硕士生, 主要研究方向为 Massive MIMO 下行传输技术及信道仿真器设计。

徐彬 (1993–), 男, 吉林吉林人, 空军航空大学工程师, 主要研究方向为信道编码、抗干扰通信。