

高误码率下 Turbo 码分量编码器快速识别算法

张立民^① 吴昭军^{*②} 钟兆根^②

^①(海军航空大学信息融合所 烟台 264001)

^②(海军航空大学电子信息工程系 烟台 264001)

摘 要: 针对目前 Turbo 码中, 分量编码器递归系统卷积码识别算法计算量大, 容错性不好两大缺点, 该文提出了一种容错性能较好的快速识别算法。首先, 在分析递归系统卷积码特殊结构的基础上, 定义了更具普遍意义的广义码重概念; 其次, 建立出递归系统卷积码生成多项式数据库, 按照数据库中多项式是否为实际编码多项式的情况, 分析出多项式所对应的结果向量广义码重概率分布; 然后, 按照分析出的广义码重概率分布, 基于极大极小准则, 导出快速识别算法判决门限的计算公式; 最后通过遍历多项式数据库, 将遍历的多项式所对应的校验方程广义码重值与判决门限比较, 从而实现参数的快速识别。仿真结果表明: 理论分析出的广义码重概率分布与仿真结果相一致, 同时算法容错性能较好, 在误码率高达 0.09 的条件下, 各种编码约束长度下的递归系统卷积码识别率在 90% 以上, 并且计算复杂度较小。

关键词: 递归系统卷积码; 多项式数据库; 判决门限; 极大极小准则; 识别

中图分类号: TN911.7

文献标识码: A

文章编号: 1009-5896(2018)01-0235-09

DOI: 10.11999/JEIT170168

A Fast Algorithm for Blind Identification of Turbo at High BER

ZHANG Limin^① WU Zhaojun^{*②} ZHONG Zhaogen^②

^①(Department of Information Fusion, the University of Naval Aeronautical, Yantai 264001, China)

^②(Department of Electronic and Information Engineering, the University of Naval Aeronautical, Yantai 264001, China)

Abstract: In order to solve the defects which are poor error tolerance and large amount of calculation in current algorithms to recognize the Recursive Systematic Convolutional (RSC) encoder in Turbo codes, a new fast algorithm is proposed. Firstly, based on special structure of RSC codes, the concept named generalized code weight is defined which is more general. Secondly, the RSC polynomial database is built up, the probability distribution of generalized code weight can be analyzed under two situation whether the polynomials in database is actual polynomial, then based on distribution and Maxmin criteria, the decision threshold of the fast algorithm is deduced. Finally, the parameters can be recognized by traversing the polynomials in database and compare the corresponding generalized code weight with decision threshold. The simulation results show that theoretical analysis of the probability distribution is consistent with the simulation's and the performance of error tolerant is preferable. The actual simulation show that correct rate of recognition can reach above 90% when the rate of bit error is as high as 0.09, besides the computational complexity is low.

Key words: Recursive Systematic Convolutional (RSC) code; Polynomial database; Decision threshold; Maxmin criteria; Identification

1 引言

为提高信息传输的可靠性, 现代通信不可避免地采用信道编码方式^[1]。目前性能较好的编码方式主

要以 Turbo 码为代表的编码方式, Turbo 码以其复杂的编码结构和优越的容错性能被广泛运用于军事通信、卫星通信、深空探测等领域^[2], 在 Turbo 码中, 递归系统卷积 (Recursive Systematic Convolutional, RSC) 码是其主要的分量编码器, 正确识别出 RSC 码的编码多项式, 对于后续交织器以及整个 Turbo 码的识别具有重要意义^[3,4]。

目前针对 RSC 码盲识别的算法主要源于传统的对卷积码的识别算法, 如解线性方程组方法求校验多项式、欧几里得算法、快速双合冲算法、Walsh-

收稿日期: 2017-02-28; 改回日期: 2017-08-17; 网络出版: 2017-09-14

*通信作者: 吴昭军 wuzhaojun1992@qq.com

基金项目: 国家自然科学基金(91538201), 泰山学者工程专项经费(ts201511020)

Foundation Items: The National Natural Science Foundation of China (91538201), The Taishan Scholar Special Foundation (ts201511020)

Hadamard 变换等^[5]。前3种算法^[6-8]主要是基于有限域中码元之间线性约束关系，但当存在误码或是误码恶化时，线性关系将遭到破坏导致识别失效，所以前3种算法虽然时效性较好但是容错性能较差；Walsh-Hadamard 变化识别算法^[9]在实现 RSC 码识别上其容错性能较强，但该算法的实质是一种穷举算法，其计算量随编码约束长度以及码元输出路数而成指数级增加；文献^[10]提出了基于校验方程符合度下的 RSC 参数识别方法，在很低的信噪比下，RSC 码编码参数的盲识别率能够达到很高，但在实际仿真中参数的初值设定比较重要，否则算法极易陷入局部极小值；此外，还有许多算法利用截获的软判决信息来识别，如文献^[11,12]，其容错性能较好但是计算量太大。综上可知，目前针对 RSC 码盲识别算法或多或少存在计算量大、容错性能差的缺点，离高性能的识别还有一定的差距。

针对目前 RSC 码盲识别算法存在的缺点，本文提出了一种高误码率条件下 RSC 快速识别算法，该算法首先建立 RSC 码生成多项式数据库，通过遍历数据库中的多项式，计算出校验结果的广义码重值，将其与设定的判决门限相比较，若大于门限值，则识别结束，反之继续遍历多项式库。与其它算法相比，本文算法计算量小，容错性高，仿真表明：在误码高达 0.09 的条件下，各种类型的 RSC 码正确识别率都在 90% 以上。

2 RSC 码编码原理

RSC 码的编码结构主要由两部分组成，即存在前向编码结构和反馈编码结构^[13]，反馈编码结构的存在使得 RSC 码的线性关联长度更长，整个结构等效于一个无限长冲激响应滤波器。图 1 是码率为 1/2 的 RSC 码编码结构示意图。图中 m_k 为 k 时刻的信息码元， c_k^1 与 c_k^2 分别为 k 时刻的第 1 路和第 2 路码元输出， D 为延时单元，生成多项式记为 $(g_{1,0}, g_{1,1}, \dots, g_{1,m}; g_{2,0}, g_{2,1}, \dots, g_{2,m})$ ，并以八进制表示，以 (7,5) 为例，生成多项式为 (1,1,1; 1,0,1)。将图 1 以编码多项式的形式来表示，可得到

$$C^1(D) = M(D) \quad (1)$$

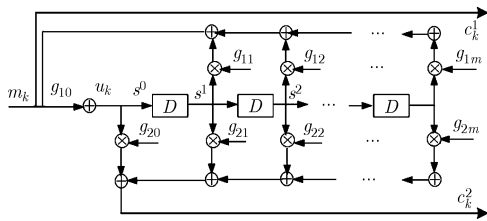


图 1 RSC 码编码结构示意图

$$C^2(D) = M(D) \cdot \frac{g_{2,0} + g_{2,1} \cdot D + \dots + g_{2,m} \cdot D^m}{g_{1,0} + g_{1,1} \cdot D + \dots + g_{1,m} \cdot D^m} \quad (2)$$

其中， $C^1(D), C^2(D)$ 为第 1 路、第 2 路的编码码元多项式，可以表示为

$$C^1(D) = c_0^1 + c_1^1 \cdot D + \dots + c_k^1 \cdot D^k + \dots \quad (3)$$

$$C^2(D) = c_0^2 + c_1^2 \cdot D + \dots + c_k^2 \cdot D^k + \dots \quad (4)$$

$M(D)$ 为编码前的信息序列，可以表示为 $m_0 + m_1 \cdot D + \dots + m_k \cdot D^k + \dots$ ，联立式(1)，式(2)可以进一步得到

$$C^2(D) \cdot (g_{1,0} + \dots + g_{1,m} \cdot D^m) \oplus C^1(D) \cdot (g_{2,0} + \dots + g_{2,m} \cdot D^m) = 0 \quad (5)$$

将截获的码元 C^1 与 C^2 ，按照式(5)在一个码元约束长度下展开，并将所得向量结果中的 0 元素映射为 1，1 元素映射为 -1，可得到

$$\mathbf{ones}(t,1) - 2 \left\{ \begin{bmatrix} c_{1,0}^2 & c_{1,1}^2 & \dots & c_{1,m}^2 \\ c_{2,1}^2 & c_{2,2}^2 & \dots & c_{2,m}^2 \\ \vdots & \vdots & \ddots & \vdots \\ c_{t,1}^2 & c_{t,2}^2 & \dots & c_{t,m}^2 \end{bmatrix} \begin{bmatrix} g_{1,m} \\ g_{1,m-1} \\ \vdots \\ g_{1,0} \end{bmatrix} \oplus \begin{bmatrix} c_{1,0}^1 & c_{1,1}^1 & \dots & c_{1,m}^1 \\ c_{2,1}^1 & c_{2,2}^1 & \dots & c_{2,m}^1 \\ \vdots & \vdots & \ddots & \vdots \\ c_{t,1}^1 & c_{t,2}^1 & \dots & c_{t,m}^1 \end{bmatrix} \begin{bmatrix} g_{2,m} \\ g_{2,m-1} \\ \vdots \\ g_{2,0} \end{bmatrix} \right\} = \mathbf{r}_{t \times 1} \quad (6)$$

$$t = \lfloor L/(m+1) \rfloor \quad (7)$$

其中， $\mathbf{ones}(t,1)$ 代表 $t \times 1$ 的列向量， L 表示截获的每一路码元数目，函数符号 $\lfloor \cdot \rfloor$ 表示向下取整。

在无误码时，式(6)中 1 元素的个数为 t 个，在截获码元中误码率一定时，当 RSC 码的前馈多项式与反馈多项式识别正确时，所得到的列向量 $\mathbf{r}$ 中 1 元素的个数一定是最大的。目前针对 RSC 码的码率识别，已有了较好的识别算法^[14,15]，在此基础上，截获的码元能够实现码元各路分离，所以本文主要的研究重点在于已知截获带有误码的 C^1 ， C^2 码元，识别 RSC 码的生成多项式。

3 基于广义码重分布下的快速识别算法

3.1 识别算法的基本思想

对于 RSC 码而言，成为 Turbo 码分量编码器的首要条件为其前馈多项式与反馈多项式之间要互素，其次必须满足使得生成 Turbo 码有效距离达到最大，而这种性能优越的 RSC 码本身数量并不大。通过计算机搜索，能够找到相应的生成多项式，同时在工程应用中，RSC 码的约束长度一般是小于等

于 7 的, 因为当约束长度增加时, 采用最大似然译码算法的复杂度急剧增加, 造成译码时延增大, 通信的时效性就得不到保障。综合以上两点可知: RSC 码的生成多项式是有限的, 文献[16]给出了约束长度 7 以内的生成多项式八进制表达式。所以对于有限的 RSC 码生成多项式, 完全可以构建出生成多项式数据库, 按照式(6)依次计算向量 $\mathbf{r}$, 那么最佳的生成多项式能够最大概率满足式(6)的约束方程。

3.2 两种情况下结果向量的广义码重分布

这里的两种情况是指从 RSC 数据库中, 调用的生成多项式是否为实际的 RSC 码编码多项式。在这两种情况下, 讨论结果向量 $\mathbf{r}_{t \times 1}$ 的广义码重分布, 首先定义向量 $\mathbf{r}_{t \times 1}$ 的广义码重:

$$W(\mathbf{r}_{t \times 1}) = \sum_{i=1}^t r_i \quad (8)$$

其中, r_i 为列向量 $\mathbf{r}$ 对应位置的元素。传统意义下的码重主要是针对 0, 1 的码元序列, 而 $\mathbf{r}_{t \times 1}$ 向量元素中包含了 1 与 -1, 定义出来的式(8)具有更加普遍的实用性。

当遍历生成多项式数据库时, 正确的多项式能够最大地满足编码约束方程, 其结果向量 $\mathbf{r}_{t \times 1}$ 中 1 元素的个数为最多, 其广义码重值一定是最大的, 即识别的多项式满足:

$$\mathbf{g}^* = \max_{\mathbf{g}} W(\mathbf{r}_{t \times 1}) \quad (9)$$

当编码多项式正确时, 结果向量 $\mathbf{r}$ 的广义码重 $W(\mathbf{r})$ 的值与所截获的码元中误码率密切相关, 从定性来看, 当无误码时, 每个线性约束编码方程都能够满足码元线性约束关系。由式(6)所得到结果向量的广义码重为 t ; 当误码情况严重时, 即误码率 $P_e = 0.5$, 每个线性约束方程只能是随机意义上成立, 结果向量 $\mathbf{r}$ 中的元素 1 与 -1 随机出现, 得到的广义码重近似为 0。下面重点定量地分析, 当多项式识别正确时, $W(\mathbf{r})$ 随误码率的分布情况。

以式(6)中某一个约束编码方程为例:

$$\mathbf{r}_i = 1 - 2 \cdot \left\{ \begin{bmatrix} g_{1,m} \\ g_{1,m-1} \\ \vdots \\ g_{1,0} \end{bmatrix} \oplus \begin{bmatrix} c_{i,0}^1, c_{i,1}^1, \dots, c_{i,m}^1 \end{bmatrix} \right\} \quad (10)$$

在误码率为 P_e , 多项式参数正确情况下, 计算 $r_i = -1$ 的概率 P 值。

分析式(10), 欲使得 $r_i = -1$, 其编码约束关系

不能够成立, 即

$$\begin{bmatrix} g_{1,m} \\ g_{1,m-1} \\ \vdots \\ g_{1,0} \end{bmatrix} \oplus \begin{bmatrix} c_{i,0}^1, c_{i,1}^1, \dots, c_{i,m}^1 \end{bmatrix} = \begin{bmatrix} g_{2,m} \\ g_{2,m-1} \\ \vdots \\ g_{2,0} \end{bmatrix} = 1 \quad (11)$$

其中, 符号 $\oplus$ 表示模 2 加。

将式(11)进一步展开, 得到:

$$\sum_{k=0}^m \oplus c_{i,k}^1 \cdot g_{2,m-k} \oplus c_{i,k}^2 \cdot g_{1,m-k} = 1 \quad (12)$$

由于误码只能出现在 $\mathbf{C}_i^1, \mathbf{C}_i^2$ 中, 分析式(12), 编码约束方程不成立必须要满足两个条件, 条件 1: 误码发生的位置只能在多项数 $\mathbf{g}_1, \mathbf{g}_2$ 参数不为 0 所对应的码元位置, 因为 0 乘以任何数都为 0, 误码将不会影响最终结果; 条件 2: 在条件 1 下的错误码元个数必须要为奇数个, 因为在二元域中, 偶数的变化不会导致结果的变化。由上述两个必要条件, 可以得到 P 的计算公式为

$$P = \sum_{k=1}^{\lfloor \frac{w+1}{2} \rfloor} \sum_{j=2k-1}^{2(m+1)} C_{2(m+1)}^j \cdot P_e^j (1 - P_e)^{2(m+1)-j} \cdot C_j^{2k-1} \left(\frac{w}{2 \cdot (m+1)} \right)^{2k-1} \left(1 - \frac{w}{2 \cdot (m+1)} \right)^{j-2k+1} \quad (13)$$

式中, w 代表生成多项式系数 $[g_2, g_1]$ 的码重, 即其中 1 元素的个数, C_i^j 表示组合数运算符。式(13)前两部分表示在一个线性约束关系下的码元序列出现了 j 个错误的概率, 后两部分表示这 j 个误码中有 $2k-1$ 码元出现在生成多项式系数为 1 的对应位置, 各种情况由求和符号连接。

当 P_e 较小时, 可以直接用 $k=1$ 的情况近似代替, 即

$$P \approx \sum_{j=1}^{2(m+1)} C_{2(m+1)}^j \cdot P_e^j (1 - P_e)^{2(m+1)-j} \cdot C_j^1 \left(\frac{w}{2 \cdot (m+1)} \right)^1 \left(1 - \frac{w}{2 \cdot (m+1)} \right)^{j-1} \quad (14)$$

求解出 P 后, $W(\mathbf{r})$ 服从均值为 $t \cdot (1 - 2P)$, 方差为 $4t \cdot P(1 - P)$ 的二项式分布, 此时遍历的多项式为正确的编码多项式。

当生成多项式不为正确的编码多项式时, 其 P 应为 0.5, 同样 $W(\mathbf{r})$ 服从二项分布, 其均值为 0, 方差为 t 。

当样本数量足够大时, 即 $t \rightarrow \infty$, 由中心极限定理可得两种情况下都应服从正态分布, 即: 当多项式为编码多项式时, $W(\mathbf{r}) \sim N(t \cdot (1 - 2P), 4t \cdot P \cdot (1 - P))$, 当多项式不为编码多项式时, $W(\mathbf{r}) \sim N(0, t)$ 。

3.3 快速识别算法的判决门限确定

门限的确定对于算法的快速识别至关重要,下面主要从3.2节所推导出的分布出发,计算判决门限值,在编码所采用的先验概率不可知的情况下,有许多准则可以求取判决门限,为了提高算法的性能,本文采用基于极大极小准则,推导判定门限。

假定遍历的多项式为正确多项式的事件为 H_1 ,事件的判决空间为 D_1 ,同时设遍历的多项式不为正确多项式的事件为 H_0 ,其判决空间为 D_0 ,在二元通信系统中,代价因子通常设定为 $c_{00} = c_{11} = 0$, $c_{10} = c_{01} = 1$,即正确判决不付出代价,错误判决代价为1,极大极小准则变换为

$$c_{10}P(D_1 | H_0) = c_{01}P(D_0 | H_1) \quad (15)$$

又因为概率密度函数 $p(x | H_0)$ 与 $p(x | H_1)$ 分别为

$$p(x | H_0) = \frac{1}{\sqrt{2\pi t}} \exp\left(-\frac{x^2}{2t}\right) \quad (16)$$

$$p(x | H_1) = \frac{1}{\sqrt{8\pi t \cdot P(1-P)}} \cdot \exp\left[\frac{-(x-t(1-2P))^2}{8t \cdot P(1-P)}\right] \quad (17)$$

设门限值为 Λ ,在极大极小准则下,应满足式(18)的积分方程:

$$\int_{\Lambda}^{+\infty} p(x | H_0) dx = \int_{-\infty}^{\Lambda} p(x | H_1) dx \quad (18)$$

设式(18)中的积分值为 α ,在标准正态分布下,概率值为 α 的下分位点为 y_{α} ,将上述两种情况下的正态分布化为标准正态分布的形式,得到式(19)与式(20):

$$\frac{\Lambda - \mu_1}{\sigma_1} = y_{\alpha} \quad (19)$$

$$\frac{\Lambda - \mu_0}{\sigma_0} = y_{1-\alpha} \quad (20)$$

式中, σ_1 与 μ_1 为第1种情况下的标准差和均值, σ_0 与 μ_0 为第2种情况下的标准差与均值,由正态分布的对称性可知, y_{α} 与 $y_{1-\alpha}$ 互为相反数,故将式(19),式(20)两式联立得到:

$$\frac{\Lambda - \mu_1}{\sigma_1} + \frac{\Lambda - \mu_0}{\sigma_0} = 0 \quad (21)$$

解得门限值 $\Lambda = \frac{\mu_1\sigma_0 + \mu_0\sigma_1}{\sigma_0 + \sigma_1}$ 。

将两种情况下推导的方差与均值代入其中,得到门限 Λ 的计算公式为

$$\Lambda = \frac{t^{3/2}(1-2P)}{t^{1/2} + (4t \cdot P(1-P))^{1/2}} \quad (22)$$

设遍历第 i 个生成多项式时,求得的门限为 Λ_i ,遍历到第 k 个生成多项式时,识别完成,则由以上的分析可得到识别概率 $P(D_1 | H_1)$ 为

$$P(D_1 | H_1) = \int_{-\infty}^{\Lambda_1} f_1(x | H_0) dx \cdots \int_{-\infty}^{\Lambda_{k-1}} f_{k-1}(x | H_0) dx \cdot \int_{\Lambda_k}^{\infty} f_k(x | H_1) dx \quad (23)$$

其中, $f_i(x | H_0)$ 为遍历的第 i 个多项式不为正确多项式时,结果向量 $\mathbf{r}$ 码重的概率密度函数, $f_i(x | H_1)$ 为遍历的第 i 正好为正确多项式时,结果向量码重的概率密度函数。

代入方差与均值于式(23),进一步得到漏警概率:

$$P(D_0 | H_1) = 1 - (\sqrt{2\pi})^{-k} \cdot \int_{-\infty}^{\Lambda_1} \cdots \int_{-\infty}^{\Lambda_{k-1}} \frac{\exp\left[-\frac{1}{2} \sum_{i=1}^{k-1} \frac{x_i^2}{\sigma_i^2}\right]}{\sigma_1 \cdots \sigma_{k-1}} dx_1 \cdots dx_{k-1} \int_{\Lambda_k}^{\infty} \frac{\exp\left[-\frac{(x - \mu)^2}{2\sigma_k^2}\right]}{\sigma_k} dx_k \quad (24)$$

设在多项式数据库中,正确的多项式编号为 z ,假定遍历到第 i 个多项式时,遍历结束,此时分两种情况讨论,求取虚警概率 $P(D_1 | H_0)$ 。

当 $i < z$ 时,遍历的多项式全部为不正确的多项式,将 i 的各种情况求和,得到此时的虚警概率为

$$P(D_1 | H_0) = \sum_{i=1}^{z-1} (\sqrt{2\pi})^{-i} \cdot \int_{-\infty}^{\Lambda_i} \cdots \int_{-\infty}^{\Lambda_{i-1}} \frac{\exp\left[-\sum_{j=1}^{i-1} \frac{x_j^2}{2\sigma_j^2}\right]}{\sigma_1 \cdots \sigma_{i-1}} dx_1 \cdots dx_{i-1} \int_{\Lambda_i}^{\infty} \frac{\exp\left[-\frac{x_i^2}{2\sigma_i^2}\right]}{\sigma_i} dx_i \quad (25)$$

当 $i \geq z$ 时,遍历的多项式库中包含正确的多项式,不妨令

$$F = (\sqrt{2\pi})^{-z} \int_{-\infty}^{\Lambda_1} \cdots \int_{-\infty}^{\Lambda_{z-1}} \frac{\exp\left[-\sum_{j=1}^{z-1} \frac{x_j^2}{2\sigma_j^2}\right]}{\sigma_1 \cdots \sigma_{z-1}} dx_1 \cdots dx_{z-1} \int_{-\infty}^{\Lambda_z} \frac{\exp\left[-\frac{(x_z - \mu)^2}{2\sigma_z^2}\right]}{\sigma_z} dx_z \quad (26)$$

此时将 i 的各种情况求和,得到虚警概率为

$$P(D_1 | H_0) = (\sqrt{2\pi})^{z-s} F \cdot \sum_{i=z+1}^s \int_{-\infty}^{A_{z+1}} \cdots \int_{-\infty}^{A_{i-1}} \frac{\exp\left(-\sum_{j=z+1}^{i-1} \frac{x_j^2}{2\sigma_j^2}\right)}{\sigma_{z+1} \cdots \sigma_{i-1}} \cdot dx_{z+1} \cdots dx_{i-1} \int_{A_i}^{\infty} \frac{\exp\left(-\frac{x_i^2}{2\sigma_i^2}\right)}{\sigma_i} dx_i \quad (27)$$

式中 s 为遍历数据库的多项式总数。

从式(24)、式(25)以及式(27)可知, 漏警概率 $P(D_0 | H_1)$ 与虚警概率 $P(D_1 | H_0)$ 不仅与方差 σ_i 和 μ 有关, 而且与遍历的多项式数目有关, 当遍历的多项式数目增加时, 漏警概率以及虚警概率都会增加。从而可知, 本文提出的识别算法, 通过筛选构建出优质的多项式库, 其虚警概率和漏警概率会大大减少, 同时识别效率也大大增加。

3.4 RSC 快速识别算法的实现

由定义的结果向量 $\mathbf{r}_{i \times 1}$ 的广义码重的意义可知, 当遍历某一个生成多项式时, 得到的 $W(\mathbf{r})$ 如果大于门限值, 则该多项式以极大的概率为实际的编码多项式, 反之则不然, 由此, 便可实现了多项式的快速识别, 避免了多项式库的全部遍历。

基于上述的分析, 得到具体的 RSC 码快速盲识别算法步骤如下:

步骤 1 依据遍历的多项式初始化截获数据, 构建数据矩阵 $\mathbf{C}_{t \times (m+1)}^1$ 与 $\mathbf{C}_{t \times (m+1)}^2$;

步骤 2 计算该遍历多项式下的结果向量广义码重值 $W(\mathbf{r})$;

步骤 3 计算最优门限值 Λ , 将步骤 2 中的广义码重值 $W(\mathbf{r})$ 与计算的门限比较, 如果 $W(\mathbf{r}) \geq \Lambda$, 输出该多项式参数, 识别结束; 反之, 重复步骤 1, 直到 $W(\mathbf{r}) \geq \Lambda$ 。

上述的识别算法虽然是针对码率为 $1/2$, 但对 $1/n$ 码率的 RSC 码识别同样适用, 因为可以将 $1/n$ 码率的 RSC 拆分成多个 $1/2$ 码率的 RSC 进行识别, 此时的计算量为 $1/2$ 码率计算量的 $n-1$ 倍。

3.5 计算复杂度分析

分析上述识别算法可知, 算法的主要计算量来源于求解广义码重值。假定截获的码元信息长度为 L , 设数据库中多项式延时单元为 m 的个数为 S_m , 工程实际 m 值大于 1 小于 7。由算法的步骤, 首先要将截获的数据构建数据矩阵 $\mathbf{C}_{t \times (m+1)}^1$ 与 $\mathbf{C}_{t \times (m+1)}^2$, 针对延时单元为 m 的多项式, 计算出广义码重 $W(\mathbf{r})$ 需要 $2t$ 次向量乘以及 t 次的数量加, 最坏的情况, 实际编码的多项式为数据库中最后一个多项式, 则得到识别一次所需要的最大向量乘法 mul 以及数量

加 sum 次数为

$$\text{mul} = \sum_{m=2}^6 2 \left\lfloor \frac{L}{m+1} \right\rfloor S_m \quad (28)$$

$$\text{sum} = \sum_{m=2}^6 \left\lfloor \frac{L}{m+1} \right\rfloor S_m \quad (29)$$

通过以上分析可知: 本文算法最大运算量仅与截获码元总数成正比, 若码率为 $1/n$, 则只需要将式(28)、式(29)乘以 $n-1$ 即可, 对于 Walsh-Hadamard 变换算法而言, 首先需要构建 $2^{2(m+1)} \times 2^{2(m+1)}$ Hadamard 矩阵, 然后在二元域中进行向量乘与数量加。设截获码长为 L , 需要二元域中的向量乘与加为

$$\text{mul} = (n-1) \left\lfloor \frac{L}{m+1} \right\rfloor 2^{2(m+1)} \quad (30)$$

$$\text{sum} = (n-1) \left\lfloor \frac{L}{m+1} \right\rfloor 2^{4(m+1)} \quad (31)$$

从式(30)与式(31)来看, Walsh-Hadamard 识别算法不仅与码元路数和截获码元长度成正比, 同时与编码约束长度成指数倍增加。

从本文算法与 Walsh-Hadamard 算法的比较来看, 本文算法通过建立 RSC 码的生成多项式库, 大大减少了不必要的数据遍历, 其计算量要远小于 Walsh-Hadamard 变换。如果在识别之前, 能够实现编码约束长度的估计, 那么算法计算量将大大减少。

4 仿真验证

4.1 两种情况下广义码重分布验证

判决门限 Λ 的值对于算法的性能有极其重要的作用, 而判决门限的推导来源于在 2 种情况下, 向量 $\mathbf{r}$ 的广义码重概率分布, 所以结果向量的广义码重概率分布的正确性验证对于整个识别算法具有重要意义。验证从均值与方差两个方面进行, 首先对比在不同误码率条件下, 结果向量 $\mathbf{r}$ 的广义码重均值的理论值与仿真值对比, 为更加有效地反映广义码重分布, 仿真设定截获的每路码元数目 $L = 10000$, 第 1 种情况下, 即遍历的多项式为实际的编码多项式时, 选择(7,5), (13,17), (25,37), (45,67), (113,133)这几种编码多项式, 误码率为 0 到 0.5, 间隔 0.02 取值, 对于第 2 种情况, 即遍历多项式不为编码多项式时, 选择(15,13)为实际编码多项式, (15,17)为遍历时的多项式; 其次, 在某一固定误码率条件下, 对比结果向量的码重分布的理论方差与仿真方差, 截获的码元长度从 500 到 10000, 间隔 500 取值, 误码率选定为 0.05, 针对某一码元长度重复仿真 100 次再求取方差。设定实际编码多项式

为(113,133), 遍历时的错误多项式为(113,147), 同样针对两种情况下, 进行对比, 得到图2所示的结果。

图2中实际的结果向量广义码重概率分布情况与理论推导的概率分布情况是比较吻合的, 从平均码重分布来看, 第1种情况下理论与仿真结果极度吻合, 而第2种情况下, 仿真结果有所波动, 但是还是在理论值0附近徘徊, 说明了在在均值这个统计量上, 理论推导出的码重分布是正确的; 从广义码重分布的方差来看, 两种情况下的仿真结果出来的方差与理论值相近, 说明从二阶统计量上来看, 理论推导的结果也是正确的。综合上述分析, 可得出, 理论推导的广义码重概率分布能够较好地反映出实际的仿真结果。

4.2 算法的有效性验证

设定截获的码元个数为5000, 误码率为0.05, 选定实际的编码多项式为约束长度为7的生成多项式(113,133), 在多项式数据库中索引号为347, 按照算法的识别步骤, 当遍历多项式数据库出现广义码重值大于计算的门限时, 识别结束, 记录了遍历不同的多项式对应的广义码重值以及出现正确多项式的门限值, 结果如图3所示, 此时的门限 $\Delta =$

152.7。

从识别的结果来看, 当遍历到347时, 第1次出现码重值大于门限, 得到了识别的生成多项式为(113,133), 估计的结果为初始设定的编码多项式, 说明算法是有效的。

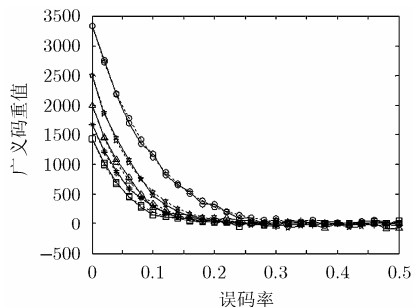
4.3 算法的容错性分析

算法的容错性能分析, 主要从编码约束长度以及截获的码元数量上来考察算法的识别性能。

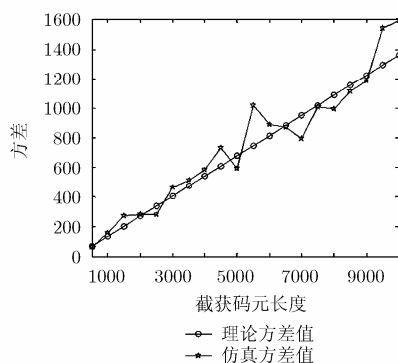
首先考察编码约束长度对算法的影响, 设定截获的码元个数为10000, 编码多项式依次为: (7,5), (13,17), (25,37), (45,67), (113,133), 代表编码约束长度从3到7, 编码结构中寄存器个数从2到6, 设定误码率为0.03到0.30, 间隔0.01取值, 仿真次数为1000次, 得到结果如图4所示。

其次考察截获码元长度对于算法识别的影响, 设定编码多项式为(25,37), 其约束长度为5, 设定截获码元数量分别为2000, 5000, 8000, 15000, 20000共5个值, 误码率从0.02到0.20, 间隔0.01取值, 蒙特卡洛仿真次数为1000次, 得到3种概率随误码率的变化情况如图5所示。

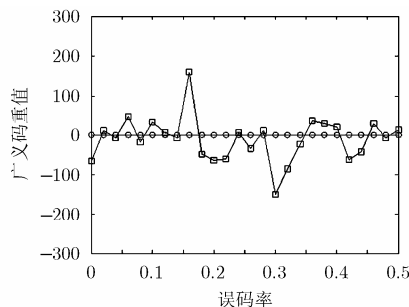
首先从图4来看, 编码约束长度对于参数的识别率影响是比较大的, 原因是当约束长度增加越大,



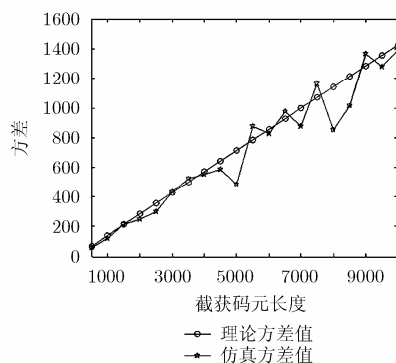
(a)第1种情况下, 广义码重均值分布对比



(c)第1种情况下, 广义码重方差对比



(b)第2种情况下, 广义码重均值分布对比



(d)第2种情况下, 广义码重方差对比

图2 两种情况下, 广义码重概率分布验证

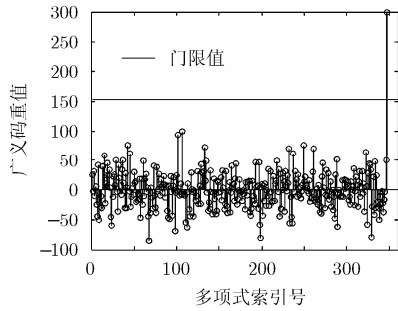


图 3 不同多项式对应的广义码重值

具有线性关系的码元出现错误概率越大, 约束关系就越容易破坏, 反映在识别率曲线上表现为随着约束长度与误码率的增加参数识别率急剧地下降; 其次从图 5 可知, 截获的码元长度能够有效地增加算法的识别性能, 原因是截获的码元越多, 越能够反映出实际的统计特性, 门限值就越精确。总地来看, 算法的容错性能是比较好的, 即使在误码率高达 0.09 时, 各类 RSC 码编码器识别率都能够达到 90% 以上。

4.4 本文算法与其他算法的比较

首先与本文算法进行比较的为容错性能较好的 Walsh-Hadamard 变换法, 比较方法是在同种条件下对比参数的识别率, 选取 4 种多项式进行识别, m 分别为 2, 3, 4, 5 得到不同误码率条件下两种算法的识别性能如图 6, 图 6 中实线为本文算法, 虚线为

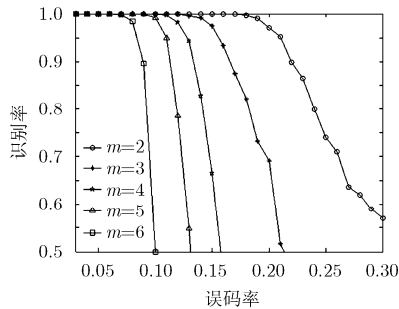


图 4 不同约束长度下 RSC 码识别率

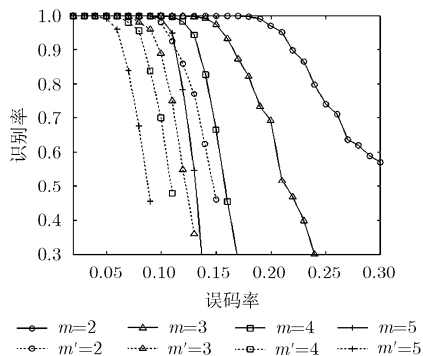


图 6 本文算法与 Walsh-Hadamard 变化对比

Walsh-Hadamard 变换。

从识别率上看, 本文算法要优于 Walsh-Hadamard 变换, 主要原因在于 Walsh-Hadamard 变换实质上是一个穷尽算法, 将各种情况都要进行遍历穷尽, 当存在误码时, 不可避免地出现虚警情况, 而本文算法构建了 RSC 码生成多项式库, 避免了 Walsh-Hadamard 变换算法无用的遍历, 一定程度上减少了虚警与漏警的情况。

其次, 与本文算法进行比较的是基于软判决的识别算法^[12]以及由 Walsh-Hadamard 识别算法改进而来的 FWHT 算法^[17], 设定调制方式为 2PSK 调制, m 为 2, 则在不同误码率条件下, 3 种识别算法识别率如图 7 所示。

从 3 种算法的对比来看, 本文提出的算法性能要远远好于文献[12]与文献[17]中的算法, 在编码约束长度为 3 的情况下, 文献[12]与文献[17]算法需要误码率小于 0.10 的条件下, 参数的识别率才能达到 90% 以上, 而本文算法在误码率高达 0.20 的条件下, 参数的识别率都能达到 90% 以上。

从运算量上进一步比较, 记录经典 Walsh-Hadamard 算法、基于软判决的识别算法以及 FWHT 算法 3 种算法, 对于不同编码约束长度的生成多项式, 识别完成一次所需要的时间, 同时与本文算法进行比较, 时间记录如表 1 所示。

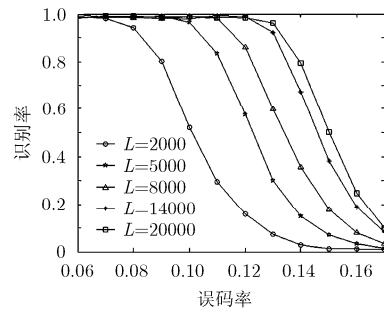


图 5 不同截获长度下 RSC 码识别率

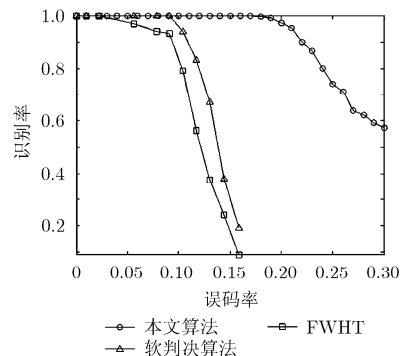


图 7 3 种识别算法的比较

表 1 4 种识别算法的运行时间对比(s)

寄存器 个数	本文 算法	Walsh-Hadamard 算法	软判决 算法	FWHT
$m=2$	0.0840	0.0182	0.7160	0.0456
$m=3$	0.0962	0.3516	1.7980	0.0485
$m=4$	0.1279	1.1859	5.1740	0.0642
$m=5$	0.2950	6.4706	21.1520	0.1440
$m=6$	1.1935	52.9980	97.6840	0.4840

从表 1 的运行时间上来看, 本文算法的时效性要远远好于 Walsh-Hadamard 算法以及软判决算法, 并且随着寄存器的个数的增加, 这种差距会更加明显; 与 FWHT 算法相比, 本文算法的时效性略差, 主要原因 FWHT 算法采用了并行的蝶形变换方式, 其运行速度大大增加, 但从图 7 来看, 本文算法的识别性能要远远优于 FWHT 算法。故本文算法要优于以上 3 种算法。

5 结束语

本文通过对 RSC 码特殊的编码结构分析, 定义了广义码重的概念。通过仿真实验, 验证了在有误差的条件下, 结果向量广义码重的理论概率分布与仿真结果是一致的, 基于极大极小准则, 推导出了能够实现快速计算的判决门限值, 从而实现 RSC 码的快速盲识别, 通过蒙特卡洛仿真验证以及与 Walsh-Hadamard 变换对比, 表明了算法容错较好并且进一步验证了算法具有较小的计算复杂度, 综合上述结论可知, 本文提出的算法容错性能好, 计算复杂度小, 具有一定的工程应用价值。

需要说明的是, 虽然本文算法适用于码率为 $1/2$ 以及码率 $1/n$ 的 RSC 码, 但对于删余型的凿孔码并不适用, 因为凿孔码通过对码元的删余, 导致信息的丢失, 使得码元之间的代数结构发生了破坏。下一步的工作是将对 RSC 码识别的算法运用于 Turbo 码的盲识别中, 探索删余型的 RSC 码的识别方法, 同时利用计算机搜索, 进一步将 RSC 码多项式数据库扩充到编码约束长度更大的情况下。

参 考 文 献

- [1] MUKHTAR H, AL-DWEIK A, and SHAMI A. Turbo product codes: applications, challenges, and future directions [J]. *IEEE Communications Surveys & Tutorials*, 2016, 18(4): 3052–3069. doi: 10.1109/COMST.2016.2587863.
- [2] LI H, GAO Z, ZHAO M, et al. Partial iterative decode of Turbo codes for on-board processing satellite platform[J]. *China Communications*, 2015, 12(11): 1–8. doi: 10.1109/CC.2015.7366233.
- [3] 任亚博, 张健, 刘以农. 高误码率下 Turbo 码交织器的恢复方法[J]. *电子与信息学报*, 2015, 37(8): 1926–1930. doi: 10.11999/JEIT141556.
- [4] REN Yabo, ZHANG Jian, and LIU Yinong. Reconstruction of Turbo-code interleaver at high bit error rate[J]. *Journal of Electronics & Information Technology*, 2015, 37(8): 1926–1930. doi: 10.11999/JEIT141556.
- [5] 刘俊, 李静, 彭华. 基于校验方程平均符合度的 Turbo 码交织器估计[J]. *电子学报*, 2016, 44(5): 1213–1217. doi: 10.3969/j.issn.0372-2112.2016.05.029.
- [6] LIU Jun, LI Jing, and PENG Hua. Estimation of Turbo-code Interleaver based on average conformity of parity-check equation[J]. *Acta Electronica Sinica*, 2016, 44(5): 1213–1217. doi: 10.3969/j.issn.0372-2112.2016.05.029.
- [7] 谢辉, 黄知涛, 王峰华. 信道编码盲识别技术研究进展[J]. *电子学报*, 2013, 41(6): 1166–1176.
- [8] XIE Hui, HUANG Zhitao, and WANG Fenghua. Research progress of blind recognition of channel coding[J]. *Acta Electronica Sinica*, 2013, 41(6): 1166–1176.
- [9] BARBIER J. Reconstruction of Turbo-code encoders[J]. *The International Society for Optical Engineer*, 2005, 5819: 463–473.
- [10] 解辉, 王峰华, 黄知涛, 等. 基于改进欧几里得算法的卷积码快速盲识别算法[J]. *国防科技大学学报*, 2012, 34(6): 159–162.
- [11] XIE Hui, WANG Fenghua, HUANG Zhitao, et al. A fast method for blind recognition of convolutional codes based on improved Euclidean algorithm[J]. *Journal of National University of Defense Technology*, 2012, 34(6): 159–162.
- [12] 邹艳, 陆佩忠. 关键方程的新推广[J]. *计算机学报*, 2006, 29(5): 711–718.
- [13] ZOU Yan and LU Peizhong. A new generalization of key equation[J]. *Chinese Journal of Computers*, 2006, 29(5): 711–718.
- [14] 刘健, 王晓军, 周希元. 基于 Walsh-Hadamard 变换的卷积码盲识别[J]. *电子与信息学报*, 2010, 32(4): 884–888. doi: 10.3724/SP.J.1146.2009.00359.
- [15] LIU Jian, WANG Xiaojun, and ZHOU Xiyuan. Blind recognition of convolutional coding based on Walsh-Hadamard transform[J]. *Journal of Electronics & Information Technology*, 2010, 32(4): 884–888. doi: 10.3724/SP.J.1146.2009.00359.
- [16] YU P D, LI J, and PENG H. A least square method for parameter estimation of RSC sub-codes of Turbo codes[J]. *IEEE Communications Letters*, 2014, 18(4): 644–647.
- [17] DEBESSU Y G, WU H C, and JIANG H. Novel blind encoder parameter estimation for Turbo codes[J]. *IEEE Communications Letters*, 2012, 16(16): 1917–1920.
- [18] 于沛东, 李静, 彭华. 一种利用软判决的信道编码识别新算法

- [J]. 电子学报, 2013, 41(2): 302-305.
- YU Peidong, LI Jing, and PENG Hua. A novel algorithm for channel coding recognition using soft-decision[J]. *Acta Electronica Sinica*, 2013, 41(5): 302-305.
- [13] 武恒州, 罗霄斌, 刘杰. Turbo 码盲识别技术研究[J]. 无线电工程, 2015, 45(5): 24-27.
- WU Hengzhou, LUO Xiaobin, and LIU Jie. Research on blind recognition of Turbo codes[J]. *Journal of Radio Engineering*, 2015, 45(5): 24-27.
- [14] 张旻, 陆凯, 李歆昊, 等. 归零 Turbo 码的盲识别方法[J]. 系统工程与电子技术, 2016, 38(6): 1424-1427. doi: 10.3969/j.issn.1001-506X.2016.06.31.
- ZHANG Min, LU Kai, LI Xinhao, et al. Blind recognition method for the Turbo codes on trellis termination[J]. *Journal of Systems Engineering and Electronics*, 2016, 38(6): 1424-1427. doi: 10.3969/j.issn.1001-506X.2016.06.31.
- [15] NASERI A, AZMON O, and FAZELI S. Blind recognition algorithm of Turbo codes for communication intelligence systems[J]. *International Journal of Computer Science Issues*, 2011, 8(6): 68-72.
- [16] 东阳. Turbo 码盲识别技术研究[实现][D]. [硕士论文], 电子科技大学, 2015.
- DONG Yang. The identification of Turbo-codes and its' implementation[D]. [Master dissertation], University of Electronic Science and Technology of China, 2015.
- [17] 林晓娴, 王维欢. SIMD-BF 模型上的并行 FWHT 算法研究[J]. 计算机时代, 2011(1): 30-32.
- LIN Xiaoxian and WANG Weihuan. A study of parallel FWHT algorithm based on SIMD-BF model[J]. *Computer Era*, 2011(1): 30-32.
- 张立民: 男, 1966 年生, 教授, 博士生导师, 主要研究方向为卫星信号处理及应用.
- 吴昭军: 男, 1992 年生, 博士生, 研究方向为信道编码识别.
- 钟兆根: 男, 1984 年生, 博士, 讲师, 主要研究方向为扩频信号处理.