

一种快速软判决译码的研究

陈 军¹, 王新梅², 曹志刚¹

(1. 清华大学电子工程系, 北京 100084; 2. 西安电子科技大学 ISN 实验室, 西安 710071)

摘 要: 本文给出一种分组码快速软判决译码—可变门限 Chase 算法 (VTC). 采用人工智能搜索技术—A* 算法, 快速生成试探序列集合, 并利用已经试探译码的信息, 对试探序列集合进行分类, 生成试探序列的等价类及其代表, 并用最优门限对候选码字进行最佳测试, 可实现快速软判决译码. 模拟计算表明, 与已有的软判决译码算法相比, 该算法的译码速度更快而译码性能完全相同.

关键词: 分组码; 软判决译码; Chase 算法; 最佳门限

中图分类号: TN911. 22 **文献标识码:** A **文章编号:** 0372-2112 (2000) 10-0074-04

Research on Fast Soft-Decision Decoding

CHEN Jun¹, WANG Xin-mei², CAO Zhi-gang¹

(1. Dept of Electronic Engineering, Tsinghua University, Beijing 100084, China;

2. ISN Lab., Xidian University, Xi'an 710071, China)

Abstract: A novel fast soft-decision decoding algorithm, called variable-threshold Chase algorithm (VTC), is described. The set of test patterns is first produced by using A* algorithm, then is partitioned into the equivalence classes. During the partition, the equivalence class representatives can be generated. A variable threshold is derived for optimality test on a candidate codeword. With set partition and optimality test, a fast soft-decision decoding algorithm is obtained. Simulation results show the advantage of the proposed algorithm over soft-decision decoding algorithms available is much faster decoding speed without loss of decoding performance.

Key words: block codes; soft-decision decoding; Chase algorithm; optimality threshold

1 引言

软判决译码有两个最佳译码准则, 一是码元错误概率最小, 如 Turbo 码的译码; 另一个是码字错误概率最小, 如 Viterbi 译码、序列译码和 Chase 译码等. 我们知道: Turbo 码的译码复杂度与译码延迟都很大, Viterbi 译码和序列译码的性能受到格图的规模以及规则性的影响, 使得这些译码方法的应用范围受到一定限制^[1,2]. Chase 译码几乎不存在这种困难问题, 译码方法相对简单, 其应用比较广泛^[3~7]. 但是只有译码复杂度低的 Chase 算法才有实用价值, 因此本文给出一个改进的 Chase 译码算法, 它采用人工智能的搜索技术—A* 算法^[8], 快速生成试探序列集合, 并利用已有试探图样的信息, 进行试探序列集合的分类和基于最佳门限的最优性测试, 进一步降低 Chase 译码复杂度, 实现接近最大似然性能的快速软判决译码. 因此该算法具有较高的实用价值.

2 软判决译码算法

假设 C 是二元 (n, k, d) 线性分组码, 其中 d 为码的最小汉明距离, 所有的码字 c 都是等概发送的. 假设传输码字为 $c_i = (c_{i1}, c_{i2}, \dots, c_{in})$. $Z = (z_1, z_2, \dots, z_n)$ 是接收序列 $r = (r_1, r_2,$

$\dots, r_n)$ 的硬判决序列, 其中 $r_i = (1 - 2c_{ii}) + n_0$, n_0 是均值为 0、方差为 σ^2 的加性高斯白噪声 (AWGN). 若 $r_i > 0$, 则 $z_i = 0$; 否则 $z_i = 1$. 硬判决译码寻找满足 $\min(w_h(e_h))$ 的码字, 其中 $e_h = (e_{h1}, e_{h2}, \dots, e_{hn}) = z - c$ 为硬判决错误图样, $w_h(e_h)$ 是 e_h 的汉明重量. 而最小欧氏距离译码意义下的最佳码字 c , 需满足 $\min d_E(c, r)$, 其中 $d_E(c, r)$ 是 r 与 c 之间的欧氏距离. Chase_(1,2,3) 算法^[3] 分别给出了三种试探序列集合, 这些集合确定了这三个算法具有不同的译码性能和译码速度. 如果 Chase 算法能够有效地利用试探序列集合, 避免产生重复的候选码字, 就会进一步提高译码速度. 这个思想是实现快速译码的关键, 首先利用该思想改进 Chase₁ 算法.

假设序列 r 的各分量已按绝对值从小到大排序: $|r_1| \leq |r_2| \leq \dots \leq |r_n|$, 即按可信度从高到低排序. 仅考虑由重量 $w = \lfloor nd/2 \rfloor$ 的 n 重试探序列 $T_j (j = 1, 2, \dots)$ 构成的试探集合 E_1 , 其中元素共有 $|E_1| = \binom{n}{w}$ 个. 令度量函数为 $f(c) = \frac{1}{4} \cdot d_E(c, r) = \frac{1}{4} \sum_{i:c_i \neq z_i} |r_i|$, 则最小欧氏距离译码也等价于寻找满足 $\min f(c)$ 的码字 c .

收稿日期: 1999-06-28; 修回日期: 2000-07-06

基金项目: 国家自然科学基金 (No. 69972035; 69772022) 和清华大学 985 基金



在译码时,试探序列 T_j 和硬判决序列 z 一起进入可纠 $t = \lfloor (d-1)/2 \rfloor$ 个错误的硬判决译码器,进行硬判决译码,译得 $|E_1|$ 个候选码字 $c_j = \text{dec}(T_j)$,选择其中具有最小 $f(c)$ 的候选码字 c_t ,作为译码输出.显然可能出现这样的情况:由某些不同的试探序列 T_j 译出的候选码字 c_j 是同一个码字.所以,如果译码算法能够避免重复的候选码字 c_t ,减少硬判决译码的次数,就能够提高译码速度.

2.1 集合分类和代表生成

对于可译的试探序列 T_j ,令 $\text{dec}(T_j)$ 表示由序列 $z + T_j$ 硬判决译码后得到的一个候选码字.如果两个可译的试探序列 T_p 和 T_q 满足 $\text{dec}(T_p) = \text{dec}(T_q)$,则 T_p 与 T_q 等价.令等价类 $Q(T_j)$ 是由与 T_j 等价的所有试探序列组成的集合,则等价类中的所有序列都对应了同一个候选码字 $\text{dec}(T_j)$.容易得到如下定理.

定理 1 对于可译的试探序列 $T_j \in E_1$,

(1) 如果 $\text{dec}(T_j)$ 与 z 的汉明距离 $d(\text{dec}(T_j), z) = t$,则 T_j 与 $\mathbf{0}$ 等价;

(2) T_j 的等价类为: $Q(T_j) = \{T_l : T_l \in E_1, \text{且 } d(T_l, z + \text{dec}(T_j)) = t\}$.

证明 因为硬判决译码器可以纠 t 个错误,所以该命题 1 成立.不失一般性,假设任意一可译试探序列 $T_l \in E_1$,与 T_j 等价: $\text{dec}(T_l) = \text{dec}(T_j)$,则有:

$$d(T_l + z, \text{dec}(T_l)) = t,$$

$$\text{即: } d(T_l, z + \text{dec}(T_l)) = d(T_l, z + \text{dec}(T_j)) = t.$$

所以, $T_l \in Q(T_j)$. 证毕.

对于任意 n 重矢量 X ,定义 $L_{X, p, q}$ 是由 X 的前 $p-1$ 个比特和后 $n-q$ 个比特被置为 0 而得到的 n 重矢量,即 $L_{X, p, q} = (0, \dots, 0, x_p, x_{p+1}, \dots, x_q, 0, \dots, 0)$.于是,可按如下方法产生等价类 $Q(T_j)$ 的代表 $\text{rep}(T_j)$:

(1) 如果 $d(z, \text{dec}(T_j)) = t$,则 $\text{rep}(T_j) = \mathbf{0}$;

(2) 如果 $d(z, \text{dec}(T_j)) > t$,则 $\text{rep}(T_j) = L_{z + \text{dec}(T_j), p, s}$,

其中 s 是矢量 $z + \text{dec}(T_j)$ 的第 w 个非零分量的比特位置.

令 E_{repr} 是由全部代表组成的集合,显然,集合 E_{repr} 大小 $|E_{\text{repr}}| \leq |E_1|$.当码长 n 比较大时, $|E_1|$ 将很大, Chase₁ 需要较多的硬判决译码,译码速度很慢,不能实用.而基于集合 E_{repr} 的译码,硬判决译码的次数相对较少,速度较快,但不损失译码性能.

令两个序列 T_p 和 $T_q \in E_{\text{repr}}$, $p = z + \text{dec}(T_p) + T_p$, $q = z + \text{dec}(T_q) + T_q$,则:

$$d(\text{dec}(T_p), \text{dec}(T_q)) = d(p - T_p, q - T_q)$$

$$d(T_q, p + T_p) + w_h(q) = d(T_q, p + T_p) + t$$

由于码的最小距离 $d \geq 2t+1$,所以有,

$$d(T_q, p + T_p) = d(T_q, z + \text{dec}(T_p)) = t + 1.$$

显然,上式的逆否命题亦成立,即有如下定理.

定理 2 若对于两个试探序列 T_p 和 T_q ,且 $T_p, T_q \in E_{\text{repr}}$,如果 T_q 满足

$$d(T_q, p + T_p) = d(T_q, z + \text{dec}(T_p)) = t \quad (2)$$

则 $T_q \in E_{\text{repr}}$.

根据定理 2,可以利用已有的试探译码信息 T_p ,检测其它试探序列 T_q 是否也是一个新的代表.如果不是代表,则丢弃这个试探序列,不对其进行试探译码,从而提高 Chase₁ 的译码速度.

2.2 集合 E_1 和 E_{repr} 的生成

遵循“优先考虑可信度较低的符号位置”的试探序列生成原则,用人工智能搜索技术——A*算法^[8],来产生所有的试探序列. A*算法保留了一个 Open 列表,其中存储了所有访问的节点和相应的路径以及路径费用.并假设该列表的容量足够大.所有的 n 重序列可用一个 $n+1$ 级的树图来表示,一条路径 $p^n = p_1 \dots p_n$ 对应唯一的一个序列,分枝 p_i 对应该序列的一个比特,分枝代价等于相应的比特值.对于级数 $l \leq n$ 的节点 m ,算法保存了从根节点到该节点的路径 $p^l(m) = p_1 \dots p_l$,和该路径的费用函数 $g(m) = p_1 + \dots + p_l$.由于利用启发信息可以提高搜索效率.因此,定义一个启发函数 $h(m)$,它等于从节点 m 到终节点的路径费用的上界估计: $h(m) = p_{l+1} + \dots + p_n$, $p_{l+1} \dots p_n$ 是一段全‘1’路径.再定义一个估价函数 $f(m) = g(m) + h(m)$.在 Open 表中, A*算法保存了已经访问过,但尚未扩展的节点,它们按费用函数从大到小的次序,在列表中从上至下排放.每一次扩展时,位于列表顶部的节点被优先扩展.节点 m 的扩展包括三个步骤:(1)产生节点 m 的两个后继节点 (m_i) , $i = 1, 2$;(2)计算后继节点的费用函数 $g(m_i)$ 和估价函数 $f(m_i)$;(3)将节点 m 从 Open 表中删除. A*算法的搜索规则为:

规则 1 如果后继节点 m_i (级数为 $l+1 \leq n$)的 $g(m_i) = w$,则算法找到了一条经过节点 m_i 且重量为 w 的路径 $p^n(m_i) = p^{l+1}(m_i) \bar{p}^{n-l-1}(m_i)$,它恰好是一个试探序列,其中,路径 $p^{l+1}(m_i) = p_1 \dots p_{l+1}$ 是已经找到的从根节点到节点 m_i 的路径, $\bar{p}^{n-l-1}(m_i) = \bar{p}_{l+2} \dots \bar{p}_n$ 是一段增补路径: $\bar{p}_i = 0, i = l+2, \dots, n$.

规则 2 如果 $g(m_i) < w$ 且 $f(m_i) \leq w$,则将节点 m_i 按排列次序插入 Open 表中.

规则 3 如果 $g(m_i) > w$ 或 $f(m_i) > w$,则丢弃该节点.

A*算法根据这三个搜索规则,进行节点扩展,最终可以找到所有汉明重量为 w 的路径,即试探序列集合.当产生了 $|E_1|$ 个试探序列时, A*算法停止.

2.3 最佳门限检测

定理 3 对于 (n, k, d) 二元线性分组码 C ,最多只有一个码字 $c_t \in C$,满足:

$$f(c) < \frac{1}{4}(d - f_h) \quad (3)$$

其中, $f_h = \sum_{i=1}^n (r_i + 2 \cdot z_i - 1)^2$,则码字 c_t 即为最大似然码字.

证明 设任意两个不同的码字 c_1 和 $c_2 \in C$, $c_1 \neq c_2$,有三角不等式

$$\{d_E(c_1, c_2)\}^{1/2} \leq \{d_E(r, c_1)\}^{1/2} + \{d_E(r, c_2)\}^{1/2}$$

两边取平方可得:

$$d_E(c_1, c_2) \leq d_E(r, c_1) + d_E(r, c_2) + 2 \{d_E(r, c_1)\}^{1/2} \cdot \{d_E(r, c_2)\}^{1/2} \leq 2\{d_E(r, c_1) + d_E(r, c_2)\}$$



由于 $d_E(c_1, c_2) \geq 4d$, 代入上式, 有:

$$d_E(r, c_1) + d_E(r, c_2) \geq 2d$$

那么在 c_1 和 c_2 中, 只有一个码字 $c_l (l=1 \text{ 或 } 2)$, 满足 $d_E(r, c_l) < d$. 该式也等价于最多只有一个码字 c_l 满足:

$$f(c) < \frac{1}{4}(d - d_E(r, z)) = \frac{1}{4}(d - f_h)$$

该码字 c_l 就是最小欧氏距离意义下的最大似然(最佳)码字. 证毕.

大量研究表明: 采用门限技术可以加快译码速度^[4-7]. 定义一个新的广义门限 $TH_{opt} = \frac{1}{4}(d - f_h)$. 显然定理 3 给出了判断候选码字 c_{tmp} 为最佳码字 c_{opt} 的充分条件. 如果新产生的候选码字满足最佳门限 TH_{opt} , 则 c_{tmp} 作为 c_{opt} 输出, 这次译码完成. 并准备接收下一个序列 r , 进行下一轮译码. 对于某一给定的接收序列 r , 最佳门限 TH_{opt} 是一个确定值. 对于不同的 r , f_h 会有改变, 所以门限 TH_{opt} 也随之变化. 因此, 该译码算法是一种可变门限 Chase₁ 译码算法 VTC₁.

2.4 对 Chase₂、Chase₃ 的改进

用和上述类似的方法, 对 Chase₂ 和 Chase₃ 也进行集合分类和最佳门限检测, 来加快译码速度, 相应地得到两个改进的译码算法—VTC₂ 和 VTC₃. 图 1 为 VTC_(1,2,3) 的流程图. 图中 * 表示硬判决译码失败.

3 与其它译码算法的比较

在译码中, 试探次数的多少以及硬判决译码速度的快慢, 都直接影响软判决译码的速度. 广义门限译码(GIC)的平均试探次数为^[5,6]: $N_{ave}(GIC) = \sum_{l=1}^{|E|} lP_l + 1$. VTC 的平均试探次数

为: $N_{ave}(VTC) = \sum_{l=1}^{|E_{rep}|} lP_l + 1$. 由于 $|E_{rep}| \leq |E|$, 因此 $N_{ave}(VTC)$

$N_{ave}(GIC)$. 此外, VTC 的度量函数 $f(c)$ 只需要简单的加法运算, 没有乘法运算, 其速度非常快, 而译码性能不受损失. 文 [7] 也采用了类似的方法, 改进 Chase₂ 算法. 与之相比, 文中给出的集合分类方法不仅简单, 而且形式更一般化, 可以适用到 Chase_(1,2,3) 三个算法中. 文 [7] 给出的最佳测试门限比较复杂, 计算很繁琐, 对不同的候选码字, 需要重新计算门限. 而 VTC 的门限 TH_{opt} 仅需要若干次算术和乘法运算, 而且不需重新计算, 因此计算更加简单.

4 模拟计算与性能分析

对 Chase₁^[3]、GIC^[5] 和 VTC 进行了模拟, 并统计了译码误比特率、试探集合大小以及译一个分组所需的硬判决译码次数. 模拟时采用了 (64, 24, 16) exBCH 线性分组码, 所得结果如图 2、3、4 所示.

图 2 给出了 VTC_(1,2,3) 的译码性能, 即误码率 P_b 与信噪比 E_b/N_0 的关系曲线. 随着试探集合的增加, VTC_(1,2,3) 的译码性能逐渐改善. 一般地, VTC₁ 的性能最好, VTC₂ 次之, VTC₃ 最差. 比如, 在 $P_b = 10^{-4}$ 时, VTC₁ 比 VTC₂ 有约 0.4dB 的译码增益, 比 VTC₃ 有 1.0dB 左右的性能增益. 在模拟时, 我们也分析了 Chase_(1,2,3) 和 GIC_(1,2,3) 的译码性能. 结果表明, Chase、GIC 以及 VTC 的三种译码性能曲线几乎完全吻合.

图 3 给出了三种分组码在不同信噪比 (E_b/N_0) 下集合 $E_{(1,2,3)}$ 和 $E_{rep(1,2,3)}$ 的平均大小. 当 E_b/N_0 增加时, $|E_{(1,2,3)}|$ 固定不变, 而 $|E_{rep(1,2,3)}|$ 有一定的减小, 而且明显地小于 $|E_{(1,2,3)}|$. 比如: 在高信噪比 ($E_b/N_0 = 7\text{dB}$) 时, $|E_{rep1}|$ 比 $|E_1|$ 小一个数量级以上, $|E_{rep2}|$ 比 $|E_2|$ 减小了 60% 以上, $|E_{rep3}|$ 比 $|E_3|$ 减小了 30%.

图 4 给出了译码算法 Chase、GIC 和 VTC 的平均试探译码次数 N_{ave} . 可以看到, 随着 E_b/N_0 的增加, Chase_(1,2,3) 的平均试探译码次数 N_{ave} 固定不变, GIC_(1,2,3) 的 N_{ave} 有一定程度的减小, 而 VTC_(1,2,3) 的 N_{ave} 不仅有明显的减小, 而且是所有曲线中相对最低的. 特别是在高信噪比时, VTC₁ 的平均试探次数 N_{ave1} 比 GIC₁ 小 1 个数量级左右, 比 Chase₁ 更小, 约 2 个数量级. VTC₂ 的平均试探次数 N_{ave2} 比 GIC₂ 减小了近一半, 比 Chase₂ 小近 1 个数量级. VTC₃ 的平均试探次数 N_{ave3} 比 GIC₃ 减小一些, 但比 Chase₃ 明显减小了 50%. 总之, VTC_(1,2,3) 的平均译码速度是最快的.

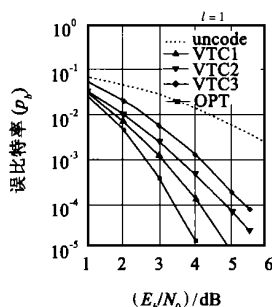


图 2 译码性能

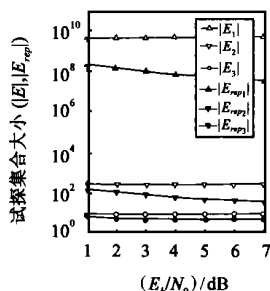


图 3 试探集合大小

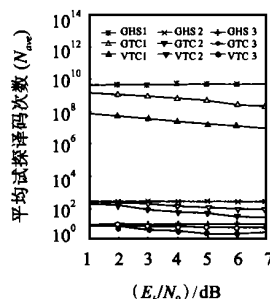


图 4 试探译码次数

综上所述, $VTC_{(1,2,3)}$ 与已有的软判决译码算法相比, 如 $Chase_{(1,2,3)}$ 和 $GIC_{(1,2,3)}$, 能够在保持相同的译码性能的同时, 显著地减少了平均试探译码次数, 大大加快了软判决译码的速度.

参考文献:

- [1] R.J. McEliece. On the BCR trellis for linear block codes [J]. IEEE Trans. Inform. Theory, July 1996, IT-42(4) : 1072 - 1092.
- [2] T. Kasami, T. Takata, T. Fujiwara, S. Lin. On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes [J]. IEEE Trans. Inform. Theory, Jan. 1993, IT-39(1) : 242 - 245.
- [3] D. Chase. A class of algorithm for linear block codes with channel measurement information [J]. IEEE Trans. Inform. Theory, Jan. 1972, IT-18(1) : 170 - 181.
- [4] G. D. Forney. Generalized minimum distance decoding [J]. IEEE Trans. Inform. Theory, Apr. 1966, IT-12(1) : 125 - 131.
- [5] 王新梅. 快速软判决译码 [J]. 电子学报, 1986, 14(4) : 64 - 71.
- [6] 马建华. 广义门限的软判决最小距离译码 [J]. 通信学报, 1990, 11(5) : 87 - 90.
- [7] H. T. Moorthy, S. Lin, T. Kasami. Soft-decision decoding of binary linear block codes based on an iterative search algorithm [J]. IEEE Trans. Inform. Theory, May 1997, IT-43(3) : 1030 - 1040.
- [8] N.J. Nilsson. Principles of Artificial Intelligence [M]. Palo Alto, CA: Tioga Publishing Co., 1980.

作者简介:



陈 军 1971 年出生, 现为清华大学电子工程系博士后. 研究兴趣包括编码与调制、数字通信系统等.

王新梅 西安电子科技大学教授, 博士生导师, 中国通信学会会士, IEEE 会员, 中国电子学会会士. 主要研究领域包括信息论、信道编码和密码学等.

曹志刚 清华大学教授, 博士生导师, 中国通信学会会士, 美国纽约科学院成员, IEEE 高级会员, 中国电子学会高级会员, 中国电子学报(英文)编委. 主要研究领域包括数据传输、编码调制、卫星通信和语音信号处理等.