

一种改进的循环冗余校验纠错技术

蒲朝飞, 张 超

(电子科技大学 电子科学技术研究院, 四川 成都 611731)

摘 要: 通常传统上使用蛮力纠错的方法来纠正接收数据中存在的随机错误. 现以 ADS-B 的 S 模式数据链为例, 分析了蛮力纠错方法在实践应用中存在的局限和不足, 并根据这些局限和不足给出了一种纠错预处理的方法, 将这种方法结合蛮力纠错方法使用, 在不改变编码的情况下大大增强了循环冗余校验码的纠错能力.

关键词: ADS-B; CRC 循环冗余码; 蛮力纠错; 纠错预处理; MATLAB

中图分类号: TN919.3

文献标识码: A

文章编号: 1000-7180(2014)04-0083-04

Modified Error Correction Using Cyclic Redundancy Code

PU Zhao-fei, ZHANG Chao

(Research Institute of Electronic Science and Technology,
University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: Traditionnally brute force technique is used to correct these random errors that receiving data usually contain. Taking the ADS-B as example, this paper analyzes shortcomings of brute force technique. In order to make up for the shortage of the brute force technique, a method of preprocessing is proposed. Both techniques enhance the error correcting ability of cyclic redundancy check.

Key words: ADS-B; Cyclic Redundancy Code; brute force technique; preprocessing; MATLAB

1 引言

信息在传输的过程中会受到噪声或干扰的影响, 为了确保所接收的数据是正确可靠的, 必须采用某种差错控制技术来对所接收的数据进行校验. 其中 CRC(Cyclic Redundancy Code)循环冗余校验在通信传输的应用十分广泛. 起初 CRC 只是用来检测出所接收数据中是否存在错误, 但是却不能将这些错误给予纠正. 当检测到错误时, 通常是采用自动请求重发(ARQ)的方式^[1]. 但是对于干扰严重的无线通信而言, 这会造成大量的数据重发, 严重地降低通信的效率. 为了使 CRC 循环冗余校验具有一定的纠错能力, 文献[2]提出了一种基于置信度判定的循环冗余纠错技术, 该纠错技术主要是利用比特置信度信息来完成纠错, 这样提高了通信效率. 但是蛮力纠错的纠错能力又很大程度上受到低置信度个数限制, 为此, 本文在分析蛮力纠错技术的优点与

不足的基础上, 提出了一种改进的纠错技术.

2 循环冗余校验^[2]

所谓循环冗余校验就是在发送端给信息码元的后面添加一些校验码, 在接收端通过这些校验码能够对前面的信息码元进行检错并予以纠正.

1090ES 数据链的 ADS-B 采用(112, 88)的循环冗余码^[3], 即在 88 bit 的数据位后面添加 24 bit 的校验位. 该循环冗余校验的生成多项式^[4]如式(1)所示:

$$g(x) = x^{24} + x^{23} + x^{22} + x^{21} + x^{20} + x^{19} + x^{18} + x^{17} + x^{16} + x^{15} + x^{14} + x^{13} + x^{12} + x^{10} + x^3 + 1 \quad (1)$$

假设在发送端发送信息码元为 $m(x)$, 那么该信息码元所对应的校验码通过 $m(x)$ 与 $g(x)$ 做模 2 求余运算得到

$$r(x) = x^{24} m(x) \bmod g(x) \quad (2)$$

得到的余数 $r(x)$ 即为校验码, 将校验码拼接信

息码后组成要传输的码字 $c(x)$.

$$c(x) = x^{24}m(x) + r(x) \quad (3)$$

在接收端, 检验传输过程中是否发生错误就是观察所接收到的数据 $s(x)$ 与 $g(x)$ 做模 2 求余运算后所得余数 $R(x)$ 是否为零.

如果在通过信道传输的过程中没有产生任何错误, 那么 $R(x) = 0$, 即:

$$\begin{aligned} R(x) &= s(x) \bmod g(x) \\ &= c(x) \bmod g(x) \\ &= \{x^{24}m(x) + r(x)\} \bmod g(x) \\ &= \{r(x) + r(x)\} \bmod 2 \\ &= 0 \end{aligned} \quad (4)$$

如果在通过信道传输的过程中产生某些错误 $e(x)$, 那么 $R(x) \neq 0$, 即:

$$\begin{aligned} R(x) &= s(x) \bmod g(x) \\ &= \{c(x) + e(x)\} \bmod g(x) \\ &= e(x) \bmod g(x) \end{aligned} \quad (5)$$

当 $R(x) \neq 0$ 时, 则说明接收到的数据一定存在错误. 当 $e(x) \neq 0$ 时, $R(x) = 0$, 则说明 $e(x)$ 为一个不可检测错误.

循环冗余码本身不具有纠错能力, 要使其具备一定的纠错能力, 必须再利用一定的置信度信息. 首先通过置信度判定机制判定的低置信度位置, 即相对于高置信度位置更容易出错的位置. 然后再利用蛮力纠错将这些位置的不同组合方式与错误校正子对比完成纠错.

3 蛮力纠错技术

我们称只存在一个错误的错误图样与 G 模 2 求余后所得余数为校正子, 那么对于 112 bit 的传输码来说, 就有 112 个校正子与之相对应. 即:

$$R_i = E_i \bmod G, 0 \leq i \leq 112 \quad (6)$$

另外我们将接收数据 S 与生成多项式模 2 求余后得到的余数 R 称为错误校正子.

设在发送端发送的正确数据为 C , 错误图样 E , 接收端接收的数据为 S . 现假设在传输过程中数据的第 8 位、第 23 位、第 77 位、第 101 位上发生了错误, 那么:

$$\begin{aligned} R &= (C + E) \bmod G \\ &= (E_8 + E_{23} + E_{77} + E_{101}) \bmod G \\ &= R_8 + R_{23} + R_{77} + R_{101} \end{aligned} \quad (7)$$

式中, ‘+’ 表示模 2 加, 从式 (7) 中可以看出错误校正子其实就等于各个错误位置所对应校正子的模 2 和.

因此我们将所有低置信度位置作对应的校正子组合后再模 2 加, 将其结果与错误校正子相对比, 如果有且仅有一组校正子组合与错误校正子相匹配, 把这个组合对应的那些低置信度位取反从而完成纠错.

假设低置信度个数为 n 时, 蛮力纠错所需要处理的组合数为 2^n , 相应未被正确接收的数量也是按 n 的指数增加, 这一特点也就决定蛮力纠错只能处理低置信度个数较少的数据, 又由于 (112, 88) CRC 循环冗余码的汉明距为 6, 所以蛮力纠错只可以纠正错误个数不大于 5 的数据.

而且蛮力纠错又要求置信度判定机制所判定的结果中, 高置信度比特位中绝对不可以包含错误比特, 低置信度比特位置中, 尽可能不包含正确比特. 换句话说低置信度位置不一定是错了的位置, 而高置信度位置一定没有错误的位置. 为了满足这一要求, 置信度判定机制在判选低置信度时会留有容限, 即通常低置信度中会存在没有错误的比特. 这样就使得蛮力纠错的纠错能力受到限制. 例如: 假设当数据中存在 5 个错误的位置, 通常低置信度的个数会大于等于 5 个, 而当低置信度个数大于 5 时, 数据将不能被纠错.

4 改进的循环冗余纠错技术

本文主要在原有的蛮力纠错的基础上通过添加纠错预处理来完成对纠错的改进. 该纠错预处理的方法可以降低低置信度的个数. 对于低置信度个数稍大于 5 的数据, 通过该纠错预处理的方法可以有效的将低置信度的个数降到 5 以下, 然后再通过原有的蛮力纠错方法对其进行纠错. 这样就使得原本不可纠正的错误变得可以纠正, 提高了系统的整体纠错能力.

4.1 纠错预处理方法

对于 112 bit 的接收数据而言, 任何一个位置上发生变化, 相应的会对错误校正子中某些特定的位置产生影响. 如图 1 所示.

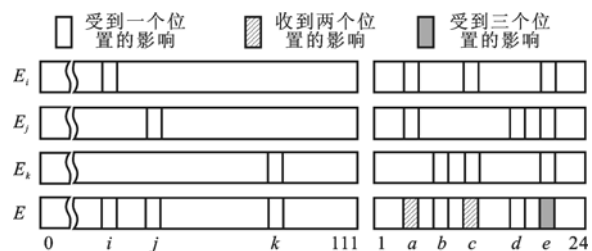


图 1 错误位置对错误校正子的影响

图中, E_i , E_j 和 E_k 分别表示在第 i 位, 第 j 位以及第 k 位上发生错误时对错误校正子产生的影响. E 表示第 i 位, 第 j 位以及第 k 位同时发生错误对错误校正子产生的综合影响.

可以看出 E 的错误校正子的第 b 位仅仅只受到数据的第 k 位的影响. 若数据的第 k 位上发生错误, 则 E 的错误校正子的第 b 位一定为 1, 若数据的第 k 位上没有发生错误, 则 E 的错误校正子的第 b 位一定为 0. 相反, 我们可以根据错误校正子第 b 位上是 0 还是 1 直接判断数据的第 k 位上是否发生了错误. 这样数据的第 k 位的置信度自然就变成高置信度. 同理可以根据错误校正子的第 d 位的情况来确定数据位第 j 位的情况. 具体的实现方法如下:

用 r_i 表示错误校正子 R 的第 i 位, 则 $R = (r_1, r_2, \dots, r_i, \dots, r_{24})$, 其中, $0 \leq i \leq 24$.

将数据 S 所有低置信位置的校正子按位做十进制加法, 把所得到的对应 24 位上的和分别表示低置信度图样对错误校正子的综合影响子. $Y = (y_1, y_2, \dots, y_i, \dots, y_{24})$; 其中 y_i 为 Y 的第 i 位, $0 \leq i \leq 24$.

如图 2 所示, 该纠错预处理方法首先检测所有 $y_i = 0$ 的位置 r_i 是否为 0. 若存在 $y_i = 0$ 且 $r_i \neq 0$ 则说明高置信度位置上存在错误, 直接将数据丢弃; 若所有 $y_i = 0$ 的位置处 r_i 均为 0, 则根据那些只受到一个低置信度位置影响的错误校正子来逐一减少低置信度的个数, 直到低置信度个数小于等于 5.

若在综合影响子的 24 位中不存在等于 1 时, 该方法将无法降低数据中低置信度的个数. 然而当低置信度个数不大于 8 时, 上述情况出现的概率并不是很高. 但随着低置信度个数的增多, 这种不可处理的情况也随之增多, 当低置信度个数为 9 时, 仅有 76% 数据可以被正确接收. 考虑到处理的效率问题, 本文只对低置信度个数不大于 8 的数据做处理.

4.2 改进的循环冗余纠错流程

纠错预处理和蛮力纠错相结合处理流程如图 3 所示.

图 3 中, 当低置信度的个数小于等于 5 的情况下直接采用蛮力纠错的方法; 当低置信度的个数大于 5 且小于等于 8 的情况下, 首先采用纠错预处理的方法来降低低置信度个数, 然后再使用蛮力纠错的方法完成纠错.

5 仿真与结果

利用 MATLAB 生成数据的低置信度图样和错误图样, 即是在 1 到 112 的范围中产生一组互不相

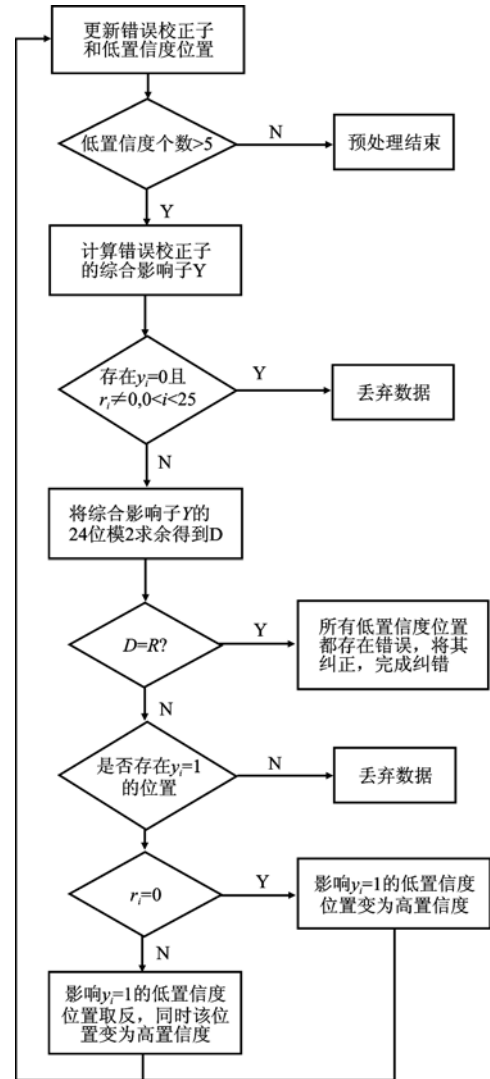


图2 纠错预处理流程图

等的 n 个随机数作为一组低置信度的位置, 在这 n 个随机数中随机选取 m 个数作为错误的位置, 其中 $m < n$. 将这些错误位置添加到 ADS-B 数据 BF2C ADA6 230D 6053 DBCC B57E EE0C 中作为发送端的发送数据. 在接收端对发送数据进行纠错并比对纠错效果.

针对每一个低置信度个数, 本实验在发送端产生 100 000 个发送数据, 这些数据中的低置信度位置和错误位置均不相同, 然后分别采用蛮力纠错以及纠错预处理和蛮力纠错相结合的方法来对这 100 000 组数据进行纠错, 将正确纠错的数据个数除以发送数据的个数作为各自的纠错能力, 其中改进前后纠错效果如表 1 所示.

从以上的仿真结果中可以看出, 在低置信度小于等于 5 的情况下, 改进前后效果一样; 但在低置信度大于 5 的情况下, 蛮力纠错是将数据完全丢失, 而

改进后的方法则正确的纠正了大部分的数据. 可见改进后的方法可以使蛮力纠错不可以纠正的错误变的可以纠正.

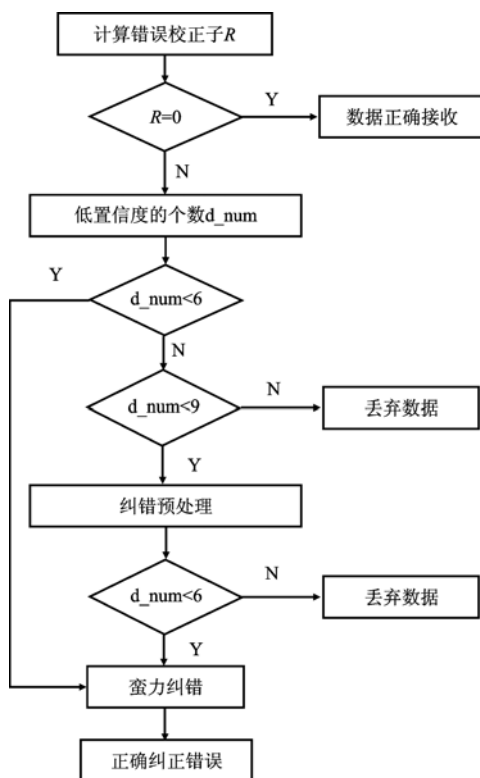


图3 纠错预处理和蛮力纠错相结合的方法

6 结束语

纠错预处理和蛮力纠错相结合的方法明显的弥补了蛮力纠错的一些不足,使原本蛮力纠错不可以纠正的错误变得可以纠正. 由于受到循环冗余的

表1 MATLAB 仿真结果

低置信度个数	发送端产生的错误数据个数	蛮力纠错正确纠正的数据个数	两种方法结合正确纠正的数据的个数
1	100 000	100 000	100 000
2	100 000	100 000	100 000
3	100 000	100 000	100 000
4	100 000	100 000	100 000
5	100 000	100 000	100 000
6	100 000	0	99 017
7	100 000	0	95 168
8	100 000	0	86 313

汉明距的影响,虽然改进后的纠错技术无法对低置信度大于5的数据予以100%纠正,但是却可以对很大一部分数据正确纠错,这无疑整体上增强了系统的纠错能力. 同时该方法对于置信度判别机制的要求相对不那么苛刻.

参考文献:

- [1] 蒋安平. 循环冗余校验码(CRC)的硬件并行实现[J]. 微电子学与计算机, 2007, 24(2): 107-110.
- [2] 陈士毅, 李玉柏. 基于置信度判定的循环冗余校验纠错技术. 电讯技术, 2007, 47(1): 175-178.
- [3] 徐赏, 黄建国, 李力. S模式数据链CRC校验实现方法[J]. 测控技术, 2012, 31(2): 27-30.
- [4] 黄晓卿. 基于FPGA的S模式信号处理[J]. 电讯技术, 2012, 52(5): 716-720.

作者简介:

蒲朝飞 男, (1988—), 硕士研究生. 研究方向为雷达信号处理、FPGA设计.