

doi:10.3969/j.issn.1001-893x.2013.01.013

RS 码类的构建及在数据链模拟系统中的应用^{*}

刘翠海^{**}, 姜 波, 温 东, 徐松岩

(海军潜艇学院, 山东 青岛 266071)

摘 要:Reed-Solomon 码(RS 码)具有强大的纠错能力,在数据通信中得到了广泛的应用,如 RS(31, 15)码已经成为美军数据链系统的标准码。为了实现应用于数据链模拟系统中的 RS 编译码类,从分析 RS 纠错编译码原理入手,在研究了 RS 编译码模型的基础上,重点对 RS 编译码的编程实现方法进行了详细分析。以 VC++ 为编程实现平台,采用面向对象的程序设计方法,构建且封装了 RS 编译码类,最后对 RS 编译码类的纠错性能进行了验证,结果表明,设计的 RS 编译码类能够有效纠错且正确恢复发端发送的信息,满足模拟系统的设计要求。

关键词:战术数据链;计算机模拟;RS 码;编译码;纠错性能

中图分类号:TN919;TP391.9 **文献标志码:**A **文章编号:**1001-893X(2013)01-0063-05

Construction of a Class for RS Code and Application in Data Link Simulation System

LIU Cui-hai, JIANG Bo, WEN Dong, XU Song-yan

(Navy Submarine Academy, Qingdao 266071, China)

Abstract:Reed-Solomon (RS) code is widely used in data communication for its strong error correction ability, such as RS (31, 15) code is adopted as a standard code in the U.S. Force's data link system. To implement the RS coding and decoding class in a data link simulation system, the RS coding and decoding principles are analyzed, and a programming method of RS code is described based on the RS coding and decoding models. With VC++ for the programming platform, the RS coding and decoding class is constructed and packaged using the object oriented programming method. Finally, the error correction performance of the RS coding and decoding class is verified, and the results show that the designed class can effectively correct the errors and recover the send information from the transmitting terminal, which meets the design requirements of the simulation system.

Key words:tactical data link; computer simulation; RS code; coding and decoding; error correction performance

1 引 言

数据链是以无线传输为主,按照统一的消息标准和通信协议,链接传感器平台、指挥控制平台和武器平台,实时处理和分发战场态势、指挥引导、战术协同、武器控制等格式化消息的系统,其本质是一种高效传输、实时分发格式化消息的信息链路^[1]。其中,严格科学的格式化消息标准是数据链广泛应用的前提条件,战术数据若不能准确传输可能会造成

提供虚假信息,致使战机贻误。为了保证战场上的各类作战数据信息准确无误地按照统一约定的格式传输分发达共享,保证作战系统的效能得到最大的发挥,在复杂的无线信道通信环境中,数据传输必须要有容错措施来增加传输数据的可靠性。由于 RS 码具有同时纠随机错误和突发错误的能力,且纠突发错误更有效,因此在数据通信和数据存储系统的差错控制中得到了广泛的应用,其中 RS(31, 15)码现已经成为美军在三军联合战术信息分发系统

^{*} 收稿日期:2012-08-01;修回日期:2012-09-24 Received date:2012-08-01;Revised date:2012-09-24
^{**} 通讯作者:qd218@sina.com Corresponding author: qd218@sina.com

(JTIDS)中采用的标准码。

本文以战术数据链通信系统为对象,详细分析了 RS 编译码模型及其编程实现方法,并采用 VC++ 语言和面向对象的程序设计方法,设计且实现了应用于数据链模拟系统中的 CRSCode 类,并针对数据传输过程中产生的突发和随机错误的极限情况进行了检验。仿真验证表明,设计的 RS 编译码类的纠错能力与理论分析一致,满足模拟系统的设计要求。

2 RS 编译码模型及编程实现

RS 码是一种多进制 BCH 循环分组码,其码字符号均在 $GF(2^m)$ 上取值。对于 (n, k) RS 码, $n = 2^m - 1$ 为码长, k 为信息码元数目, $n - k$ 为监督码元数目, $d = n - k + 1$ 表示码元距离。RS 码是一种极大最小距离可分码,即在所有线性分组码中其最小距离最大,也就意味着 RS 码(纠 $2t = n - k$ 个差错)的纠错能力最强^[4-5]。

RS 码的基本思想就是选择一合适的多项式 $g(x)$, 并且使得对每个信息码计算得到的码字多项式 $c(x)$ 都是 $g(x)$ 的倍式,即使得 $c(x)$ 除以 $g(x)$ 的余数多项式为 0。这样,如果接收到的码字多项式除以 $g(x)$ 的余数多项式不为 0,则可以知道接收的码字中存在错误,因此可以进一步计算并纠正 t 个错误。

2.1 RS 编译码模型及编程实现

RS 码属于循环码的一种,它的编码过程就是利用其信息多项式 $m(x)$ 乘以 x^{n-k} 后再除以生成多项式 $g(x)$, 所得余式即为校验多项式 $h(x)$, 然后将 $h(x)$ 置于 $m(x)$ 之后,即形成 RS 码。其中,各多项式的所有运算均在伽罗华域进行,因此,在计算机仿真实验中,有必要首先完成伽罗华域元素产生的程序。

2.1.1 构造伽罗华域 $GF(2^m)$ ^[2]

对于 $GF(2^m)$ 中的元素,首先给定 $1 = \alpha^0$, 连乘 α 可以在 m 重的形式里通过对 α^0 连续左移来产生 α^1 、 α^2 、 $\cdots$ 、 α^{m-1} 。对 α^{m-1} 的左移生成 $10\cdots 0$ (共 m 个 0), 左移结果该元素为 $m+1$ 比特,因此必须要映射到 m 比特,即通过 $\alpha^m = \alpha + 1$ 多项式。其比特数据计算方法是 将 $m+1$ 重数中的最高位掩去得到 m 重数后,通过按比特的模 2 加“11”到掩码后的 m 重结果(即加 $\alpha + 1$)来实现的。随后继续左移可以产生 α^{m+1} 、 α^{m+2} 、 $\cdots$ 、 α^{2m-1} 。伽罗华域元素产生编程实现的框图如图 1 所示。

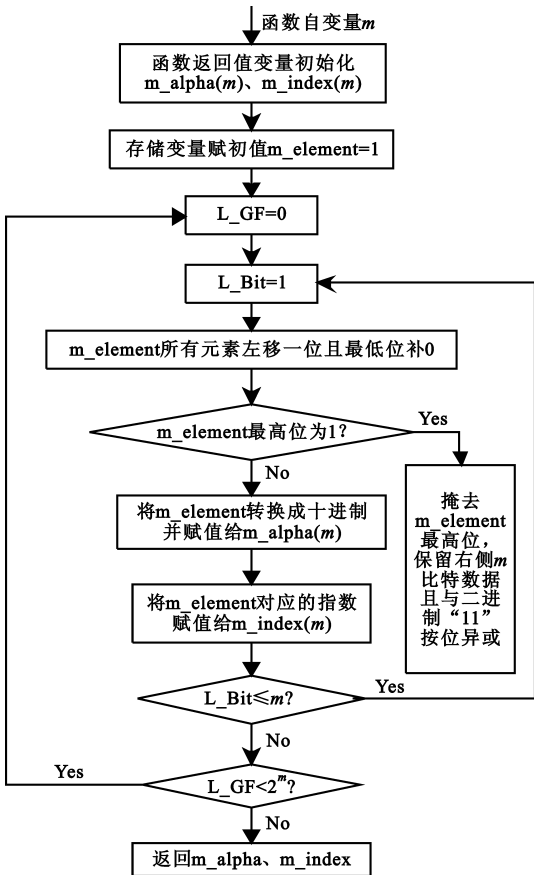


图 1 伽罗华域元素产生流程图
Fig.1 Generation of a Galois Field's element

2.1.2 多项式的计算机表示及数学运算模型

有限域 $GF(p^m)$ 中的每个元素均可由一个多达 m 项的多项式表示,每一项的系数是一个模 p 值。在计算机中,系数可以存储在一个数组里,其中 x^n 的系数存放到数组中的位置 n ,即假设数组从 0 开始标记,如 C 和 C++。对于像 Matlab 从 1 开始标记的语言, x^n 的系数将存到数组中 $n+1$ 的位置。对于 $GF(2^m)$ 中的元素,多项式可以存储在一个整型变量中,用每比特代表一个二进制系数。

RS 编码、译码过程中用到了大量的多项式元素的乘法、加法和倒数运算。我们知道,其运算均是在伽罗华域 $GF(2^m)$ 中进行的,其乘法运算对应的是指数模 2^m 相加,加法运算则是模 2 运算。为此,在实现了伽罗华域元素产生函数的基础上,需要编程实现 RS 编译码中常用的多项式加法、乘法、除法等基本运算函数。

假设有两个多项式

$$A(x) = a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0$$
$$B(x) = b_r x^r + b_{r-1} x^{r-1} + \cdots + b_1 x + b_0$$

多项式加法运算实现起来非常简单,多项式乘法与

除法编程实现基本相同,图 2 给出了多项式乘法计算电路。

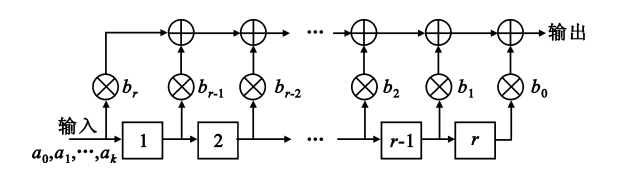


图 2 多项式乘法电路

Fig.2 Circuit of multiplicity of the polynomial

按照图 2 所示电路多项式乘法的实现步骤如下^[6]:

- 步骤 1: r 个寄存器全部清零;
- 步骤 2: $A(x)$ 最高次系数 a_k 首先送入时,乘法器输出乘积的最高次项 x^{k+r} 的系数 $a_k b_r$,同时 a_k 存入寄存器的第一级;
- 步骤 3: $A(x)$ 的第二个系数 a_{k-1} 送入时, a_k 由第一级寄存器进入第二级寄存器,同时与 b_{r-1} 相乘,乘法器输出 $a_{k-1} b_r + a_k b_{r-1}$,即得到了乘积的 x^{k+r-1} 的系数;
- 步骤 4: 如此重复进行,直至 $k+r+1$ 次移位后,乘法器输出乘积的常数项 $a_0 b_0$ 。

2.1.3 RS 编码的编程实现

假设需发送的信息码字序列为 $m_1, m_2, \dots, m_k$, RS 码的编码电路如图 3 所示。

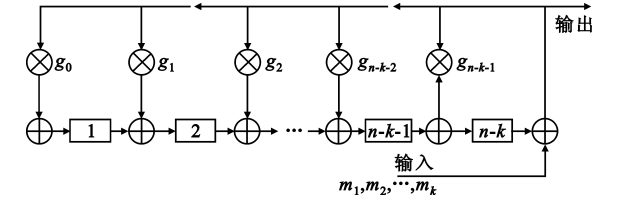


图 3 RS 编码电路

Fig.3 Encoding circuit for RS code

按照图 3 所示电路实现的 RS 编码计算机程序流程如图 4 所示。

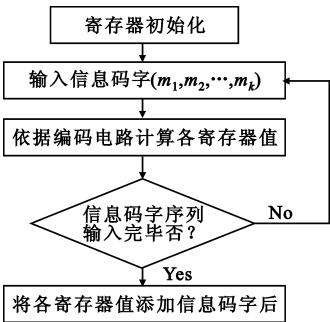


图 4 RS 编码程序流程图

Fig.4 Flowchart of encoding program of RS code

2.2 RS 译码模型及编程实现^[6]

相对于 RS 编码,其译码过程比较繁琐、复杂。RS 译码主要包括计算伴随多项式、确定差错位置多项式、计算差错位置多项式的根和求错误图样 4 个步骤。第一、三、四步计算机编程实现起来比较简单,第一步求伴随多项式是将 $\alpha^i (i = 1, 2, \dots, 2t)$ 代入接收多项式 $r(x)$ 而得到的;第三步依据钱氏搜索算法确定错误位置,得到差错数量;第四步根据错误位置和错误值得到错误图样。第二步是整个译码过程中最复杂的一步,大多采用 BM 迭代算法。实现 BM 迭代算法可以以列表形式完成,如表 1 所示,表中 l_u 是 $\sigma^{(u)}(x)$ 的次数。

表 1 BM 迭代过程

Table 1 BM algorithm

u	$\sigma^{(u)}(x)$	d_u	l_u	$u - l_u$
-1	1	1	0	-1
0	1	s_1	0	0
1				
2				
$\vdots$				
$2t$				

假设我们已经填满了第 u 行,可以这样来填第 $u+1$ 行:

- (1)如果 $d_u = 0$,则 $\sigma^{(u+1)}(x) = \sigma^{(u)}(x)$ 且 $l_{u+1} = l_u$;
- (2)如果 $d_u \neq 0$,寻找第 u 行前的第 ρ 行,使 $d_\rho \neq 0$ 及最后一列 $(\rho - l_\rho)$ 为最大值,则由式(1)可得 $\sigma^{(u+1)}(x)$,且
$$\sigma^{(u+1)}(x) = \sigma^{(u)}(x) - d_u d_\rho^{-1} x^{(u-\rho)} \sigma^{(\rho)}(x) \quad (1)$$
$$l_{u+1} = \max(l_u, l_\rho + u - \rho) \quad (2)$$
- (3)不管 d_u 是否等于零,都要依据式(3)计算 d_{u+1} :

$$d_{u+1} = s_{u+2} + \sigma_1^{(u+1)} s_{u+1} + \dots + \sigma_{l_{u+1}}^{(u+1)} s_{u+2-l_{u+1}} \quad (3)$$

式中, $\sigma_i^{(u+1)}$ 是 $\sigma^{(u+1)}(x)$ 的系数。

最后一行的多项式 $\sigma^{(2t)}(x)$ 即为需要的 $\sigma(x)$ 。如果它的次数大于 t ,则表明接收序列的错误多于 t ,一般来说就无法决定它们的位置。

3 RS 码类的构建和在模拟系统中的实现

数据链模拟系统以战术数据链通信系统为模拟对象,以 VC++ 语言为编程实现平台,采用面向对象的程序设计方法,以战术数据链通信系统各模块为类对象,编程实现了各类对象并进行了封装。基于 RS 编译码的数据链模拟系统结构如图 5 所示。

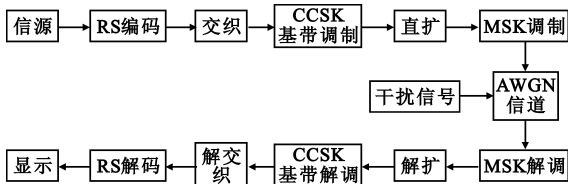


图 5 数据链模拟系统信号流程图

Fig.5 Block diagram of the data link simulating system

依据图 5 各模块的划分,数据链模拟系统设计并实现了 CModel 类,该类是 VC++ 中 CObject 类的派生类,定义了供所有模型类调用的公共过程,是图 5 所有模块的基类。CRSCode 类、CCSK 类、CMSK 类以及 CDSSS 类都是 CModel 类的派生类,它们分别是 RS 编译码类、循环码移位键控类、最小移频键控类、直接序列扩频类的基类。

3.1 RS 编译码类的整体设计

根据上一节的分析,CRSCode 类定义有自己的成员变量和成员函数,完成整个模拟系统中的编码和译码。下面给出了 CRSCode 类定义的部分内容:

```
class CRSCode:public CModel
{
public:
    int n;//码字长度
    int k;//消息字长度
    int m;//伽罗华域各元素的比特数
    int generate_GF (int m);//生成 GF(2^m)域所有元素
    int multiply_rs(int a, int b, int m);
    int reverse_rs(int a, int m);
    int polymul_rs (int a,int b,int m);
    int polydiv_rs(int a,int b,int m);
}
```

CRSCode 类中的成员函数 multiply_rs(int a, int b, int m),参数 a、b 为十进制表示的多项式中 x^n 的系数,参数 m 为 $GF(2^m)$ 域中各元素的比特数;函数返回值为系数 a 与 b 相乘的结果。

成员函数 reverse_rs(int a, int m),参数 a 为十进制表示的多项式中 x^n 的系数,参数 m 为 $GF(2^m)$ 域中各元素的比特数;函数返回值为系数 a^{-1} 的结果。

成员函数 polymul_rs (int a,int b,int m),参数 a、b 为一数组,是以十进制表示的多项式中从 $x^0 \sim x^n$ 各项的系数,a、b 的排列顺序是以多项式次数从低到高的顺序排列,参数 m 为 $GF(2^m)$ 域中各元素的比特数;函数返回值为系数多项式 $a(x)$ 与 $b(x)$ 相乘的结果。

成员函数 polydiv_rs(int a,int b,int m),参数 a、b 为一数组,是以十进制表示的多项式中从 $x^0 \sim$

x^n 各项的系数,a、b 的排列顺序是以多项式次数从低到高的顺序排列。参数 a 为待发送信息码字,参数 b 为生成多项式,参数 m 为 $GF(2^m)$ 域中各元素的比特数;函数返回值为校验多项式各系数,排列顺序是以多项式次数从低到高的顺序排列。

3.2 RS 编译码类在模拟系统中的应用及校验

数据链模拟系统模拟了时分多址的工作方式,每个时隙长为 7.812 5 ms。在不考虑同步信息的情况下,传输的消息包括信息标志和数据两部分^[3]。信息标志部分主要包括了时隙号、用户号、消息类型等报头信息。数据部分模拟了固定格式的消息字,每个消息字包含 15 个码元,每个码元 5 b 字符,其中数据占 70 b,奇偶校验位占 5 b。为保证传输消息的可靠性,信息标志部分采用的是 RS(16,7)编码,对消息字采用的是 RS(31,15)编码。通过 (16,7)RS 编码,在传输中即使这 7 个码元有 4 个码元出现了错误也可得到纠正。数据部分的 (31,15)RS 编码,每组可纠 8 个错误,7 组交织可纠 56 个错误。这种编码使得在由 31 个字符组成的大组中,由于干扰或传播等原因,即便有 16 个字符无法判决或有 8 个字符判决错误,在译码时也能得到纠正。

为了检验 CRSCode 类的运算速度和纠错能力及正确性,分别对数据传输过程中产生的突发和随机错误的极限情况进行了检验,表 2 给出了信源产生 10^4 个符号、信噪比在 0~7 dB 范围测得的实验数据,根据数据绘出的性能曲线如图 6 所示。仿真结果表明 CRSCode 类的纠错能力与理论分析一致,与设计的功能相符,达到了译码的预期效果。

表 2 误码率实验测量数据

Table 2 BER data of test

E_b/N_o	BER
0.0	0.080 00
0.5	0.075 00
1.0	0.063 10
1.5	0.050 20
2.0	0.041 30
2.5	0.361 00
3.0	0.029 00
3.5	0.023 10
4.0	0.181 00
4.5	0.131 00
5.0	0.009 20
5.5	0.005 11
6.0	0.001 22
6.5	0.000 21
7.0	2.1×10^{-6}

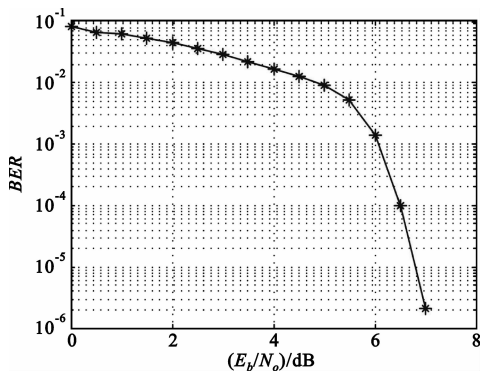


图 6 RS 编码 BER 性能曲线

Fig.6 Reed-Solomon error performance evaluation

4 结束语

数据链模拟系统的实现大体可分为建立模型、用建立的模型构造目标系统和系统仿真三部分工作,构造系统的模型是完成所有工作的基础。由于 RS 编译码在战术数据链通信系统中有着重要的意义,它能够提高整个系统的可靠性和抗干扰能力,进而提高系统性能,因而准确地对其建模、编程实现及封装是整个数据链模拟系统中关键的环节。本文以 VC++ 为软件平台,以面向对象的程序设计方法实现了 RS 编译码类,并简要介绍了数据链模拟系统的程序组成。本模拟系统只考虑了编码、交织、CC-SK 基带调制以及 MSK 调制等模型,没有考虑同步以及其他信道模型,因此还需要进一步完善。

参考文献:

[1] 刘翠海. 美军战术数据链的发展及作战运用[J]. 电讯技术, 2007,47(5):6-10.
LIU Cui-hai. The Development and Operational Application of the U. S. Forces' Tactical Dada Links[J]. Telecommunication Engineering, 2007,47(5):6-10. (in Chinese)

[2] 王昕. 无线通信系统仿真 - C++ 实用模型[M]. 北京: 电子工业出版社,2005.

WANG Xin. Simulating Wireless Communication Systems Practical Models in C++ [M]. Beijing: Publishing House of Electronics Industry, 2005. (in Chinese)

[3] 梅文华, 蔡善法. JTIDS/Link16 数据链[M]. 北京: 国防工业出版社,2007.
MEI Wen-hua, CAI Shan-fa. JTIDS/Link16 Data Link[M]. Beijing: National Defense Industry Press,2007. (in Chinese)

[4] 陶德元, 何小海, 吴志华. RS 码编译码算法的实现[J]. 四川大学学报(自然科学版),2000,37(6):868-872.
TAO De-yuan, HE Xiao-hai, WU Zhi-hua. The Implementation of RS Encoding and Decoding[J]. Journal of Sichuan University(Natural Science Edition), 2000,37(6):868-872. (in Chinese)

[5] 韩作生, 袁东风. RS 码频域编译码的计算机模拟[J]. 通信学报,1994,15(6):104-112.
HAN Zuo-sheng, YUAN Dong-feng. Computer Simulation of the Coding and Decoding of RS Code in Frequency Domain [J]. Journal of China Institute of Communications, 1994, 15 (6):104-112. (in Chinese)

[6] 杨爵, 郎宗棣. 实用纠错编码[M]. 北京: 中国铁道出版社,1988.
YANG Jue, LANG Zong-yan. Practical Error Correcting Coding[M]. Beijing: Chinese Railroad Publishing House, 1988. (in Chinese)

作者简介:



刘翠海(1963—),男,河北唐山人,硕士,副教授,主要研究方向为军事通信及仿真;
LIU Cui-hai was born in Tangshan, Hebei Province, in 1963. He is now an associate professor with the M. S. degree. His research concerns military communication and simulation.
Email: qd218@sina.com

姜波(1978—),男,江苏新沂人,博士,讲师,主要研究方向为无线通信和数字信号处理。
JIANG Bo was born in Xinyi, Jiangsu Province, in 1978. He is now a lecturer with the Ph. D. degree. His research concerns wireless communication and digital signal processing.
Email: bojiang@nudt.edu.cn