



**哈尔滨工业大学（深圳）**  
Harbin Institute of Technology, Shenzhen

2022 年 春 季学期课程设计报告

考 核 科 目 : 通信工程方法论

报 告 题 目 : 基于 MATLAB 的三种信道编码  
方式通过不同信道的仿真对比

学生所在院（系） : 电子与信息工程学院

组 员 姓 名 : 通信四班 200210409 陆豪

通信二班 200210226 陈子祁

通信二班 200210232 侯任宣

上 交 日 期 : 2022. 06. 19

考 核 结 果 :

审 阅 人 :

## 摘 要

本论文以探究汉明码、循环码、卷积码的编码译码方式及其在 AWGN 信道、瑞利信道、莱斯信道中误码率的指标为目的，采用控制变量、理论与仿真相结合的研究方法，研究了不同编码方式与不同信道之间结合的三种误码率对比方式，基于 MATLAB 的仿真全面分析了图像的趋势与现象的本质原因，在已选的编码方式与信道中总结出误码率最低、性能最优质的编码方式及其信道，并进行归纳总结得出对应的结论及其规律。

关键词：汉明码；循环码；卷积码；AWGN 信道；瑞利信道；莱斯信道；误码率

## Abstract

The purpose of this paper is to explore the coding and decoding methods of Hamming codes, cyclic codes and convolutional codes and their bit error rate indicators in AWGN channels, Rayleigh channels and Rice channels. The research method is to study the three comparison methods of bit error rate combined between different coding methods and different channels. The simulation based on MATLAB comprehensively analyzes the trend of the image and the essential reasons of the phenomenon, and summarizes the selected coding methods and channels. The coding method and its channel with the lowest bit error rate and the best performance are summarized, and the corresponding conclusions and laws are obtained.

**Keywords:** Hamming code, cyclic code, convolutional code, AWGN channel, Rayleigh channel, Rice channel, bit error rate.

# 目 录

摘 要.....	- 2 -
第 1 章 绪 论.....	- 6 -
1.1 课题背景及研究的目的和意义.....	- 6 -
1.2 信道编码及其相关理论的概述.....	- 7 -
1.2.1 信道编码的基本原理.....	- 7 -
1.2.2 信道编码的分类.....	- 7 -
1.3 本文的主要研究内容.....	- 8 -
第 2 章 汉明码的理论与编码译码方式.....	- 9 -
2.1 线性分组码的基本原理.....	- 9 -
2.2 线性分组码的编码理论.....	- 9 -
2.3 线性分组码的距离与纠检错能力.....	- 11 -
2.4 线性分组码的译码理论.....	- 12 -
2.5 汉明码的编码译码过程.....	- 13 -
第 3 章 循环码的理论与编码译码方式.....	- 16 -
3.1 引言.....	- 16 -
3.2 (N,K)循环码.....	- 17 -
3.2.1 (n,k)循环码概念.....	- 17 -
3.2.2 循环码编码原理.....	- 18 -
3.3 循环码译码.....	- 19 -
3.3.1 循环码纠错原理.....	- 19 -
3.3.2 编码过程及实现.....	- 21 -
3.3.3 译码过程及实现.....	- 22 -
第 4 章 卷积码的理论与编码译码方式.....	- 25 -
4.1 引言.....	- 25 -
4.2 编码原理.....	- 25 -
4.2.1 编码器结构.....	- 25 -
4.2.2 状态图.....	- 27 -
4.2.3 篱笆图（网格图）.....	- 28 -
4.3 维特比（VITERBI）算法译码.....	- 29 -
第 5 章 仿真结果与理论分析.....	- 33 -
5.1 对于同一编码方式不同信道之间的误码率曲线关系.....	- 34 -
5.1.1 对于汉明码编码方式不同信道之间的误码率曲线关系.....	- 34 -
5.1.2 对于循环码编码方式不同信道之间的误码率曲线关系.....	- 35 -
5.1.3 对于卷积码编码方式不同信道之间的误码率曲线关系.....	- 36 -

5.1.3.1 对于(3,1,2)卷积码编码方式不同信道之间的误码率曲线关系.....	- 36 -
5.1.3.2 对于(2,1,2)卷积码编码方式不同信道之间的误码率曲线关系.....	- 38 -
5.1.3.3 对于(2,1,4)卷积码编码方式不同信道之间的误码率曲线关系.....	- 39 -
5.2 对于同一信道不同编码方式之间的误码率曲线关系.....	- 42 -
5.2.1 对于 AWGN 信道中不同编码方式的误码率曲线关系.....	- 42 -
5.2.2 对于瑞利信道中不同编码方式的误码率曲线关系.....	- 43 -
5.2.3 对于莱斯信道中不同编码方式的误码率曲线关系.....	- 44 -
5.3 对于瑞利信道下不同种卷积码之间的误码率曲线关系.....	- 45 -
结 论.....	- 47 -
参考文献.....	- 48 -

# 第1章 绪 论

## 1.1 课题背景及研究的目的和意义

随着现代无线通信技术的迅猛发展，数字信号已经逐渐取代了模拟信号成为主要的传输信号类型。与模拟信号相比较，数字通信具有高抗干扰能力，易于加密提高保密性，可以使用现代数字信号处理技术等优势。但由于信道的特性复杂，当调制好的信号在信道里进行传输的时候，必然要受到信道的影响。信道的影响可以分成以下三个主要方面：第一是信道本身对信号产生的衰落：由于信道本身频率响应特性不理想，造成对信号的破坏；第二是信道中的各种噪声，如背景噪声，脉冲噪声等等，这些噪声叠加在信号上面，改变信号的幅度、相位和频率，使信号在解调时产生错误；第三，是信号在传输过程中由于反射，折射或沿不同路径传播从而带来的叠加效应，即通常所说的多径效应，这会带来时间上前后信号互相干扰。总而言之，这三种加性干扰都会导致在接收端信号解调的错误，使系统的误码率大大增加。导致其在无线信道中传输过程中极易受到干扰而使码元波形变坏，所以传输到接收端后可能发生错误判决。而信道条件越是恶劣，产生错误的可能性就越大。

由于信道中的加性干扰不能通过均衡等方法完全消除，因此在一个实用的通信系统中，必须采取一定的措施来纠正错误，降低系统的误码率。信道编码就是一种非常有效的措施。信道编码的任务就是，在发送端以可控的方式在信号中加入一定的冗余度，而在接收端这些冗余度可以用来检测并且纠正信号通过信道后产生的错误。当然，冗余度的加入降低了系统的工作效率，但是和系统误码率的降低（即信号更加正确地传送）相比，这些代价是可以接受的。信道编码正是基于此提出并发展起来的。

## 1.2 信道编码及其相关理论的概述

### 1.2.1 信道编码的基本原理

信道编码，就是在待发送的信息序列中人为地按一定规则加入保护成分（监督码元），然后将构成的新序列作为发送序列；接收端的信道译码器按相应逆规则进行译码，从中发现错误或纠正错误。从信息传输的角度来看，监督码元是冗余的，这种冗余度降低了信息传输的效率；但从系统可靠性角度来看，冗余度的增加带来了检（纠）错能力的提升，增强了数字信号的抗干扰能力。因此，信道编码又称差错控制编码。

设编码后的发送序列共包含 $n$ 个码元，其中信息码元数为 $k$ ，监督码元数为 $(n-k)$ 。信息码元数与总码元数的比值 $k/n$ 称为编码效率（码率），监督码元数与信息码元数的比值 $(n-k)/k$ 称为冗余度。一般来说，冗余度越大，信息抗干扰能力越强，系统可靠性越高。信道编码的本质即通过降低信息传输速率（有效性）来提高信息的抗干扰能力（可靠性）。

### 1.2.2 信道编码的分类

信道编码一般分为几大类，其中常见的包含：分组编码和卷积编码。

在分组编码中，信息序列被划分为若干个长度为 $k$ 的码组，每个信息码组按一定的编码规则映射成长度为 $n$ 的发送码组。各码组的映射关系是独立的，编码器的输出仅与当前输入的 $k$ 个信息码元有关，而与先前的序列无关。对于 $(n-k)$ 分组码，码率定义为 $k/n$ 。在卷积编码中，虽然也将长度为 $k$ 的信息码组编成长度为 $n$ 的发送码组，但其监督位不仅和当前输入的 $k$ 个信息码元有关，同时也与前面 $(N-1)$ 个信息码组有关，故一个码组中的监督码元监督着 $N$ 个信息码组。对于 $(n,k,N)$ 卷积码，称 $N$ 为编码约束度，码率定义为 $k/n$ 。

### 1.3 本文的主要研究内容

本论文以探究汉明码、循环码、卷积码的编码译码方式及其在 AWGN 信道、瑞利信道、莱斯信道中误码率的指标为目的，采用控制变量、理论与仿真相结合的研究方法，研究了不同编码方式与不同信道之间结合的三种误码率对比方式，基于 matlab 的仿真全面分析了图像的趋势与现象的本质原因，在已选的编码方式与信道中总结出误码率最低、性能最优质的编码方式及其信道，并进行归纳总结得出对应的结论及其规律。



## 第 2 章 汉明码的理论与编码译码方式

### 2.1 线性分组码的基本原理

分组码的基本思想是对信息序列进行分组编码。对包含  $k$  个码元的码组  $M = (m_{k-1}, m_{k-2}, \dots, m_1, m_0)$  按照一定的编码规则产生包含  $n$  个码元的码组  $C = (c_{n-1}, c_{n-2}, \dots, c_1, c_0)$ ，编码规则定义为：

$$\begin{cases} c_0 = f_0(m_{k-1}, m_{k-2}, \dots, m_1, m_0) \\ c_1 = f_1(m_{k-1}, m_{k-2}, \dots, m_1, m_0) \\ \dots \\ c_{n-1} = f_{n-1}(m_{k-1}, m_{k-2}, \dots, m_1, m_0) \end{cases}$$

若  $f_i(\cdot) (i=0, 1, \dots, n-1)$  均为线性函数，则称  $C$  为线性分组码，一般用  $(n, k)$  表示，其中  $n$  表示码组长度， $k$  为码组中信息码元长度， $r = n - k$  为码组中监督码元长度。

实际上， $(n, k)$  线性分组码是  $q$  元有限域  $GF(q)$  上  $n$  维线性空间  $V_n$  中的一个  $k$  维子空间  $V_{n,k}$ 。若信息码组  $M$  与码组  $C$  的所有元素均取自二元有限域  $GF(2)$ （即  $\{0, 1\}$ ），则称为二元线性分组码，接下来讨论二元码的情况。

### 2.2 线性分组码的编码理论

对于二元线性分组码，其编码过程实际上就是从包含  $2^k$  个信息码组的  $V_k$  空间到包含  $2^n$  个码组的  $V_n$  空间的映射过程，因此在码空间  $V_{n,k}$  中一定可以找到一组基底  $g_0, g_1, \dots, g_{k-1}$ ，使得所有码组都可以写成这  $k$  个基底的线性组合，故称为线性分组码，即

$$C = m_{k-1}g_{k-1} + m_{k-2}g_{k-2} + \dots + m_1g_1 + m_0g_0$$

用  $g_i = (g_{i,n-1}, g_{i,n-2}, \dots, g_{i,1}, g_{i,0})$  表示第  $i (i=0, 1, \dots, k-1)$  个基底，再将  $k$  个基底排列成  $k$  行  $n$  列矩阵的形式，即

$$G = \begin{bmatrix} g_{k-1} \\ \vdots \\ g_1 \\ g_0 \end{bmatrix} = \begin{bmatrix} g_{k-1,n-1} & \cdots & g_{k-1,1} & g_{k-1,0} \\ \vdots & \ddots & \vdots & \vdots \\ g_{1,n-1} & \cdots & g_{1,1} & g_{1,0} \\ g_{0,n-1} & \cdots & g_{0,1} & g_{0,0} \end{bmatrix}$$

由于  $k$  个基底，即  $G$  的  $k$  个行向量线性无关，故矩阵  $G$  的秩一定等于  $k$ 。当信息码组  $M$  确定后，码组  $C$  仅由  $G$  矩阵决定，即  $(n, k)$  线性分组码中的任一码组  $C$  均可由这组基底的线性组合生成：

$$C = [m_{k-1}, m_{k-2}, \cdots, m_0] \begin{bmatrix} g_{k-1} \\ \vdots \\ g_1 \\ g_0 \end{bmatrix} = MG$$

因此称这  $k \times n$  矩阵  $G$  为该  $(n, k)$  线性分组码的生成矩阵。例如，可以在  $(7, 4)$  线性分组码中找到任意 4 个线性无关的行向量来构成生成矩阵  $G$ ：

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

由于基底不是唯一的，生成矩阵也就不是唯一的。事实上，将  $k$  个基底线性组合后产生另一组  $k$  个向量，只要满足线性无关的条件，依然可以作为基底张成另一个码空间。不同的基底也有可能生成同一个码组，但因编码涉及码组和映射两个因素，即使码组相同而映射方法不同也不能认为是同样的码。

基底的线性组合等效于生产  $G$  矩阵的行运算，能够产生一组新的基底。利用矩阵的行运算可使生产矩阵具有如下的“系统形式”：

$$G = [I_k \ P] = \begin{bmatrix} 1 & 0 & \cdots & 0 & p_{k-1,n-k-1} & \cdots & p_{k-1,1} & p_{k-1,0} \\ 0 & 1 & \cdots & 0 & p_{k-2,n-k-1} & \cdots & p_{k-2,1} & p_{k-2,0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & p_{0,n-k-1} & \cdots & p_{0,1} & p_{0,0} \end{bmatrix}$$

其中  $P$  为  $k \times (n-k)$  矩阵， $I_k$  为  $k \times k$  单位矩阵，从而保证矩阵  $G$  的秩为  $k$ 。对于系统码，有

$$C = MG = \begin{bmatrix} M & MP \end{bmatrix}$$

根据线性代数知识，生成  $G$  矩阵是由  $k$  个线性无关的行向量构成的，因此一定存在一个由  $n-k$  个线性无关的行向量构成的矩阵  $H$  与之相交，即  $GH^T = 0$ 。对于“系统形式”的生成矩阵  $G$ ，其校验矩阵  $H$  也是规则的：

$$H = \begin{bmatrix} -P^T & I_{n-k} \end{bmatrix}$$

对于二条码，由于模二减法等同于模二加法，上式的负号可以省略。

## 2.3 线性分组码的距离与纠检错能力

在分组码中，把码组中“1”的数目称为码组的重量，简称码重；把两个码组中对应位置上数字不同的位数称为码组的距离，简称码距，又称汉明距离，记为  $d$ 。某种编码算法中各个码组之间距离的最小值称为最小码距：

$$d_0 = \min \{d_{C_i C_j}, i \neq j, C_i, C_j \in V_{n,k}\}$$

两个码组之间的距离表示它们之间差别的大小：距离越大，两个码组的差别越大，传输时从一个码组错成另一码组的可能性越小，抗干扰能力越强。估算最小码距是纠错码设计的重要步骤，原始方案是逐一计算两两码组之间的距离，找到其中的最小值；然而若每个码集有  $2^k$  个许用码组，就需要计算  $2^k(2^k - 1)/2$  个距离，计算量太大。利用线性分组码的封闭性：任意两个码组之和仍为许用码组，即

$$C_i + C_j = C_m \in C$$

因此任意两个码组之间的码距就是另一码组的码重，表达式如下：

$$d_{C_i C_j} = w(C_i + C_j) = w(C_m)$$

式中  $w(C_m)$  表示码组  $C_m$  的码重。最小码距为  $d_0$  可表示为：

$$d_0 = \min \{w(C_m), C_m \in V_{n,k}, C_m \neq 0\}$$

将计算最小码距问题转化成寻找最轻码字问题，若每个码集有  $2^k$  个许用码组，仅需计算  $2^k$  次。

对于最小码距为  $d_0$  的线性分组码，其检错能力  $e = d_0 - 1$ ，纠错能力：

$$t = \left\lfloor \frac{d_0 - 1}{2} \right\rfloor$$

式中 $\lfloor \cdot \rfloor$ 表示向下取整。

码的纠错能力取决于码的最小距离，但还需说明的另一点是码的总体纠错能力不仅仅与 $d_0$ 有关。纠错能力 $t$ 只是说明距离 $t$ 以内的差错一定能纠正，并非说距离大于 $t$ 的差错一定不能纠正。事实上，如果每个码集有 $2^k$ 个许用码组，就存在 $2^k(2^k - 1)/2$ 个距离，且每个距离并非是相等的。比如最小距离 $d_0 = 3$ ，纠错能力 $t = 1$ ：若许用码组 $C_2$ 与 $C_1$ 之间的距离等于最小距离 3，而 $C_2$ 与 $C_3$ 之间的距离大于最小距离 3（此处假设为 5）；只要 $C_2$ 朝 $C_1$ 方向的偏差大于 1 就会出现译码错误，然而若 $C_2$ 朝 $C_3$ 方向偏差 2，译码时仍可正确地判断为 $C_2$ 而非 $C_3$ 。可见，总体的、平均的纠错能力不但与最小距离有关，而且与其余码距离或者说与码组的重量分布特性有关，把码距（码重）的分布特性称为距离（重量）谱，其中最小的重量就是 $d_0$ 。正如信息论各符号等概时熵最大一样，从概念上可以想象到：当所有码距相等时（重量谱为线谱）码的性能应该最好；或者退一步说，当各码距相当不大时（重量谱为窄谱）性能应该称得上好。事实证明确实如此，在同样的 $d_0$ 条件下，窄谱的码一般比宽谱的码更优。

纠错重量谱的研究具有理论与现实意义，不仅仅是计算各种译码差错概率的主要依据，也是研究码的结构、改善码集内部关系从而发现新的好码的重要工具。但目前除了少数几类码如汉明码、极长码等的重量分布已知外，还有很多码的重量分布并不知道，距离分布与性能之间确切的定量关系对于大部分码而言尚在进一步研究当中，特别当 $n$ 和 $k$ 较大时，要得出码重分布是非常困难的。

## 2.4 线性分组码的译码理论

编码后输出的码组在信道上传输时可能产生一定的码元错误，将这些码元错误称为错误图样。设发送码组为 $C = (c_{n-1}, c_{n-2}, \dots, c_1, c_0)$ ，信道产生的错误图样为

$E = (e_{n-1}, e_{n-2}, \dots, e_1, e_0)$ ，则接收到的码组为  $R = C + E = (r_{n-1}, r_{n-2}, \dots, r_1, r_0)$ ，译码器的作用就是根据接收到的码组  $R$  来估计错误图样  $E$ ，进而得到对发送码组  $C$  的估计。

对于二源码，模二加法等同于模二减法，故错误图样  $E = R - C = R + C$ 。利用码组与校验矩阵  $H$  的正交性即  $CH^T = MGH^T = 0$ ，可检验接收码组  $R$  是否出错：

$$RH^T = (C + E)H^T = CH^T + EH^T = EH^T \begin{cases} = 0 \\ \neq 0 \end{cases}$$

定义伴随式  $S = RH^T = EH^T$ ，由上式可得伴随式  $S$  仅与错误图样  $E$  有关，而与发送码组  $C$  无关：若在信道传输过程中没有错误发生即  $E = 0$ ，则  $S = 0$ ；否则  $S \neq 0$ 。伴随式译码算法的基本思想就是根据伴随式  $S$  的值来估计错误图样  $E$ 。将校验矩阵  $H$  写成列向量的形式：

$$H = [h_{n-1}, h_{n-2}, \dots, h_1, h_0]$$

则有

$$S = EH^T = \sum_{i=0}^{n-1} e_i h_i^T$$

即伴随式  $S$  是校验矩阵  $H$  中列向量的线性组合。对于错误图样  $E$ ，码元中第  $j$  位发生错误时其值  $e_j = 1$ ，否则  $e_j = 0$ 。因此伴随式  $S$  的值实际上是出错码元对应的校验矩阵  $H$  的列向量的模二和。在线性分组码的纠错能力范围内，如果能够确定伴随式  $S$  的值是校验矩阵  $H$  的哪个或哪几个列向量的模二和，就可以确定错误图样  $E$ ，进而实现译码。

## 2.5 汉明码的编码译码过程

汉明码是最常见的线性分组码，最小距离  $d_0 = 3$ ，纠错能力  $t = 1$ ，即能够纠正单个随机错误。以  $(7, 4)$  汉明码为例，生成矩阵  $G$  如下：

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

根据  $GH^T = 0$ ，校验矩阵  $H$  为：

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

根据  $C = MG = [M \quad MP]$ ，信息码元  $M$  与监督码元  $MP$  对应关系如表 1 所示。

表 1 信息码元和监督码元对应关系

信息码元 ( $c_6c_5c_4c_3$ )	监督码元 ( $c_2c_1c_0$ )	信息码元 ( $c_6c_5c_4c_3$ )	监督码元 ( $c_2c_1c_0$ )
0000	000	1000	111
0001	011	1001	100
0010	101	1010	010
0011	110	1011	001
0100	110	1100	001
0101	101	1101	010
0110	011	1110	100
0111	000	1111	111

根据  $S = RH^T = EH^T$ ，伴随式  $S$  与错误图样  $E$  的关系如表 2 所示。

表 2 伴随式与错误图样对应关系

伴随式 ( $s_2s_1s_0$ )	错误图样 ( $e_6e_5e_4e_3e_2e_1e_0$ )	伴随式 ( $s_2s_1s_0$ )	错误图样 ( $e_6e_5e_4e_3e_2e_1e_0$ )
000	0000000	011	0001000
001	0000001	101	0010000
010	0000010	110	0100000
100	0000100	111	1000000

设输入信息序列  $M = (1 \ 0 \ 0 \ 1)$ ，按表 1 进行编码可得：

$$C = MG = [1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0]$$

得到编码输出

$$C = [1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0]。$$

设信号在传输时发生错误，得到译码输入序列  $R = [1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]$ ，计算

伴随式  $S = RH^T = [0 \ 1 \ 1]$ 。根据表 2 可得错误图样  $E = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]$ ，即接收序列第四位发生错误，纠正后得到正确的发送序列为：

$$C = R + E = [1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0]$$

可知信息序列  $M = [1 \ 0 \ 0 \ 1]$

## 第3章 循环码的理论与编码译码方式

### 3.1 引言

循环码是线性分组码的一种，所以它具有线性分组码的一般特性，此外还具有循环性。循环码的编码和解码设备都不太复杂，且纠错能力强。它不但可以检测随机的错误，还可以检错突发的错误。 $(n, k)$  循环码可以检测长为  $n-k$  或更短的任何突发错误，包括首尾相接突发错误。循环码是一种无权码，循环码编排的特点是相邻两个数码之间满足卡诺图中的邻接条件，即相邻两个数码之间只有一位码元不同，码元就是组成数码的单元。符合这个特点的有多种方案，但循环码只能是表中的那种。循环码的优点是没有瞬时错误，因为在数码变换过程中，在速度上会有快有慢，中间经过其它一些数码形式，称它们为瞬时错误。这在某些数字系统中是不允许的，为此希望相邻两个数码之间仅有一位码元不同，即满足邻接条件，这样就不会产生瞬时错误。循环码就是这样一种编码，它可以在卡诺图中依次循环得到。循环码又称格雷码。循环码最大的特点就是码字的循环特性，所谓循环特性是指：循环码中任一组许用码组经过循环移位后，所得到的码组仍然是许用码组。若  $( )$  为一循环码组，则  $( )$ 、 $( )$  ……还是许用码组。

循环码对码有一些附加结构约束。它们都是基于伽罗华域，由于其结构性质，循环码对差错控制很有用。它们与伽罗华域密切相关，因此编码和译码算法都方便计算。



## 3.2 (n,k)循环码

### 3.2.1 (n,k)循环码概念

在实际应用中，数据传输一般采用系统码的编码方式，即在发送的信息序列之后附加上特定位数序列的冗余位，该冗余位称为所发送的信息序列的监督位。监督位一般是由所发送的信息序列经过恰当的变化而产生。若监督位由信息序列经过线性组合得到，则称得到的系统码为线性分组码。

循环码是线性分组码的一个重要子类，具有严密的代数学理论。循环码“线性”是指任意两个循环码模 2 相加所得的新码仍为循环码。循环码具有线性码的一般性质（即封闭性，指一种线性分组码的任意两个码组之和仍是该分组码的另一个码组）外，还具有循环性，即循环码中任一码组循环一位（将最右端码元移至左端，或反之）以后，仍为该码组中的一个码组。 $(n, k)$  循环码表示其中信息位为  $k$ ，监督位为  $n-k$  位。

若一个循环码的所有码字多项式都是一个次数最低的非零首一多项式  $g(x)$  的倍式，则  $g(x)$  生成该码，并称  $g(x)$  为该码的生成元或生成多项式。若在  $GF(2)$  上的  $(n, k)$  循环码中，存在唯一的  $n-k$  次首一多项式  $g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \dots + g_1x + g_0$  使得每一个码多项式  $c(x)$  都是  $g(x)$  的倍式，且每一低于或等于  $n-1$  次的  $g(x)$  的倍式，一定是码多项式。 $(n, k)$  循环码的生成多项式  $g(x)$  一定是  $x^n + 1$  的因式： $x^n + 1 = g(x)h(x)$ ；反之，若  $g(x)$  为  $n-k$  次，且除尽  $x^n + 1$ ，则此  $g(x)$  一定生成一个  $(n, k)$  循环码。

对于一般的  $(n, k)$  循环码，设其生成多项式为  $g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \dots + g_1x + g_0$ ，由于  $g(x)$ 、 $xg(x)$ 、 $\dots$ 、 $x^{k-1}g(x)$  等  $k$  个码多项式必线性无关，故可用它们组成码的一组基底，而与这些码多项式相对应的  $k$  个线性无关的码向量就构成除生成矩阵  $G$ ，即

$$G(x) = \begin{bmatrix} g_{n-k} & g_{n-k-1} & \dots & g_1 & g_0 & 0 & \dots & 0 \\ 0 & g_{n-k} & g_{n-k-1} & \dots & g_1 & g_0 & 0 & \dots & 0 \\ \dots & & & & & & & & \\ 0 & \dots & 0 & g_{n-k} & g_{n-k-1} & \dots & g_1 & g_0 \end{bmatrix}$$

若  $g(x)$  是  $(n, k)$  循环码的生成多项式，有  $x^n + 1 = g(x)h(x)$ ， $h(x)$  是  $k$  次多项式，称为校验多项式。令  $h(x) = h_k x^k + h_{k-1} x^{k-1} + \dots + h_1 x + h_0$ ，则

$$H = \begin{bmatrix} h_0 & h_1 & \dots & h_k & 0 & \dots & 0 \\ 0 & h_0 & h_1 & h_2 & \dots & h_k & \dots & 0 \\ \dots & \dots & & & & & & \\ 0 & \dots & & 0 & h_0 & h_1 & \dots & h_k \end{bmatrix}$$

为  $(n-k) \times n$  阶矩阵，称为码的校验矩阵。可以证明， $G \cdot H^T = 0$ 。

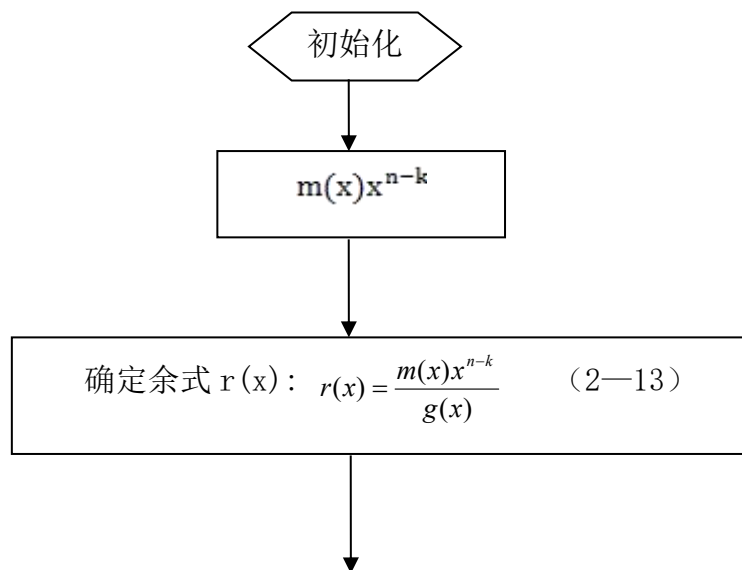
### 3.2.2 循环码编码原理

有信息码构成信息多项式  $m(x) = m_{k-1} x^{k-1} + \dots + m_0$ ，其中最高幂次为  $k-1$ ；

用  $x^{n-k}$  乘以信息多项式  $m(x)$ ，得到的  $x^{n-k} m(x)$ ，最高幂次为  $n-1$ ，该过程相当于把信息码  $(m_{k-1}, m_{k-2}, \dots, m_1, m_0)$  移位到了码字德前  $k$  个信息位，其后是  $r$  个全为零的监督位；

用  $g(x)$  除  $x^{n-k} m(x)$  得到余式  $r(x)$ ，其次数必小于  $g(x)$  的次数，即小于  $(n-k)$ ，将此  $r(x)$  加于信息位后做监督位，即将  $r(x)$  于  $x^{n-k} m(x)$  相加，得到的多项式必为一码多项式。

根据上面的讨论，可得到在  $(7, 4)$  循环码编码的程序框图如图 1 所示：



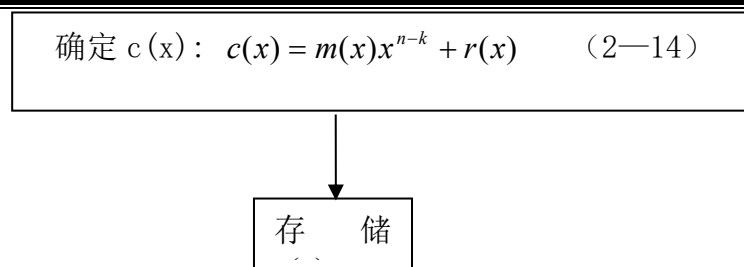


图 3.2.1 编码程序框图

### 3.3 循环码译码

#### 3.3.1 循环码纠错原理

纠错码的译码是该编码能否得到实际应用的关键所在。译码器往往比编码较难实现，对于纠错能力强的纠错码更复杂。根据不同的纠错或检错目的，循环码译码器可分为用于纠错目的和用于检错目的的循环码译码器。

通常，将接收到的循环码组进行除法运算，如果除尽，则说明正确传输；如果未除尽，则在寄存器中的内容就是错误图样，根据错误图样可以确定一种逻辑，来确定差错的位置，从而达到纠错的目的。用于纠错目的的循环码的译码算法比较复杂，感兴趣的话可以参考一些参考书。而用于检错目的的循环码，一般使用 ARQ 通信方式。检测过程也是将接受到的码组进行除法运算，如果除尽，则说明传输无误；如果未除尽，则表明传输出现差错，要求发送端重发。用于这种目的的循环码经常被成为循环冗余校验码，即 CRC 校验码。CRC 校验码由于编码电路、检错电路简单且易于实现，因此得到广泛的应用。在通过 MODEM 传输文件的协议如 ZMODEM、XMODEM 协议中均用到了 CRC 校验技术。在磁盘、光盘介质存储技术中也使用该方法。

当码字  $c$  通过噪声信道传送时，会受到干扰而产生错误。如果信道产生的错误图样是  $e$ ，译码器收到的  $n$  重接受矢量是  $y$ ，则表示为：

$$y = c + e \quad (1-1)$$

上式也可以写成多项式形式：

$$y(x) = c(x) + e(x) \quad (1-2)$$

译码器的任务就是从  $y(x)$  中得到  $e(x)$ ，然后求的估值码字

$$\hat{c}(x) = y(x) + e(x) \quad (1-3)$$

并从中得到信息组  $\hat{m}(x)$ 。

循环码译码可按以下三个步骤进行：

(1) 有接收到的  $y(x)$  计算伴随式  $s(x)$ ；

(2) 根据伴随式  $s(x)$  找出对应的估值错误图样  $\hat{e}(x)$ ；

(3) 计算  $\hat{c}(x) = y(x) + \hat{e}(x)$ ，得到估计码字  $\hat{c}(x)$ 。若  $\hat{c}(x) = c(x)$ ，则译码正确，否则，若  $\hat{c}(x) \neq c(x)$ ，则译码错误。

由于  $g(x)$  的次数为  $n - k$  次， $g(x)$  除  $E(x)$  后得余式（即伴随式）的最高次数为  $n-k-1$  次，故  $S(x)$  共有  $2^{n-k}$  个可能的表达式，每一个表达式对应一个错误格式。可以知道 (7, 4) 循环码的  $S(x)$  共有  $2^{(7-4)} = 8$  个可能的表达式，可根据错误图样表来纠正 (7, 4) 循环码中的一位错误，其伴随式如表 1 所示。

BCH (7, 4) 循环码错误图样表：

错误图样	错误图样码字	伴随式 $S(x)$	伴随式
$E_6(x)=x_6$	1000000	$x^2$	100
$E_5(x)=x_5$	0100000	$x^2+x$	110
$E_4(x)=x_4$	0010000	$x^2+x+1$	111
$E_3(x)=x_3$	0001000	$x+1$	011
$E_2(x)=x_2$	0000100	$x^2+1$	101
$E_1(x)=x_1$	0000010	$x$	010
$E_0(x)=x_0$	0000001	1	001
$E(x)=0$	0000000	0	000

表 3.3.1 BCH (7, 4) 循环码错误图

上式指出了系统循环码的译码方法：将收到的码字 $R(x)$ 用 $g(x)$ 去除，如果除尽则无错；否则有错。如果有错，可由余式 $S(x)$ 一一找出对应图样，然后将错误图样 $E(x)$ 与 $R(x)$ 模2和，即为所求码字 $C(x)$ ，从而实现纠错目的。

根据前面的讨论，可得（7，4）循环码译码的程序框图如图 2.3.1 所示：

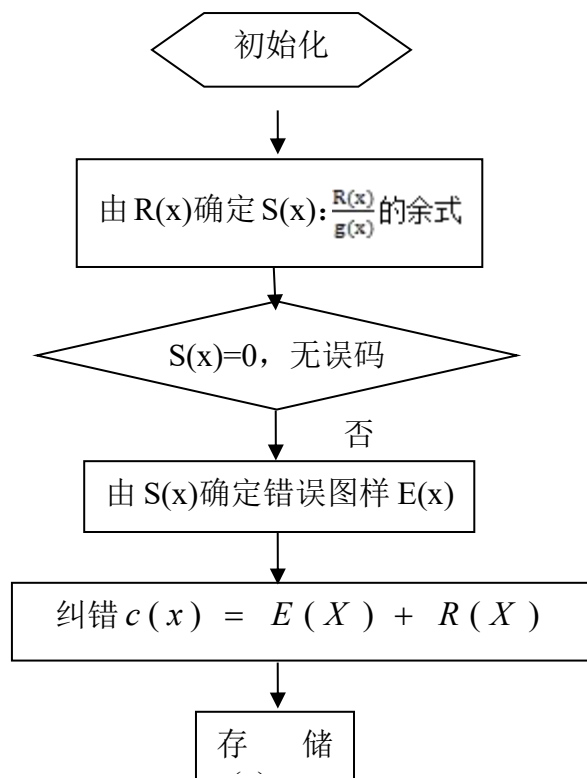


图 3.3.2 译码程序框图

### 3.3.2 编码过程及实现

在编码时，首先需要根据给定循环码的参数确定生成多项式 $g(x)$ ，也就是从 $x^n+1$ 的因子中选一个 $(n-k)$ 次多项式作为 $g(x)$ ；然后，利用循环码的编码特点，即所有循环码多项式 $A(x)$ 都可以被 $g(x)$ 整除，来定义生成多项式 $g(x)$ 。

根据上述原理可以得到一个较简单的系统：设要产生 $(n, k)$ 循环码， $m(x)$ 表示信息多项式，循环码编码方法则其次数必小于 $k$ ，而 $x^{n-k} \cdot m(x)$ 的次数必小于 $n$ ，用 $x^{n-k} \cdot m(x)$ 除以 $g(x)$ ，可得余数 $r(x)$ ， $r(x)$ 的次数必小于 $(n-k)$ ，将 $r(x)$ 加

到信息位后作监督位，就得到了系统循环码。下面就将以上各步处理加以解释：

(1) 用  $x^{n-k}$  乘  $m(x)$ 。这一运算实际上是把信息码后附加上  $(n-k)$  个“0”。例如，信息码为 110，它相当于  $m(x) = x^2 + x$ 。当  $n-k=7-3=4$  时， $x^{n-k} \cdot m(x) = x^6 + x^5$ ，它相当于 1100000。而希望的到得系统循环码多项式应当是  $A(x) = x^{n-k} \cdot m(x) + r(x)$ 。

(2) 求  $r(x)$ 。由于循环码多项式  $A(x)$  都可以被  $g(x)$  整除，也就是：

$$\frac{A(x)}{g(x)} = Q(x) = \frac{x^{n-m} \cdot m(x) + r(x)}{g(x)} = \frac{x^{n-m} \cdot m(x)}{g(x)} + \frac{r(x)}{g(x)} \quad (1-4)$$

因此，用  $x^{n-k} \cdot m(x)$  除以  $g(x)$ ，就得到商  $Q(x)$  和余式  $r(x)$ ，即

$$\frac{x^{n-m} \cdot m(x)}{g(x)} = G(x) - \frac{r(x)}{g(x)} \quad (1-5)$$

这样就得到了  $r(x)$ 。

(3) 编码输出系统循环码多项式  $A(x)$  为： $A(x) = x^{n-k} \cdot m(x) + r(x)$  (1-6)

### 3.3.3 译码过程及实现

对于接收端译码的要求通常有两个：检错与纠错。达到检错目的的译码十分简单，可以由式 (1)，通过判断接收到的码组多项式  $B(x)$  是否能被生成多项式  $g(x)$  整除作为依据。当传输中未发生错误时，也就是接收的码组与发送的码组相同，即  $A(x)=B(x)$ ，则接收的码组  $B(x)$  必能被  $g(x)$  整除；若传输中发生了错误，则  $A(x) \neq B(x)$ ， $B(x)$  不能被  $g(x)$  整除。因此，可以根据余项是否为零来判断码组中是否有错码。

设发射的码字为  $C(x)$ ，而收到的码字为  $R(x)$ ，如果  $C(x) = R(x)$ ，则说明收到码字正确。如果  $C(x) \neq R(x)$ ，则说明收到的码字出现错误，即为： $R(x) = C(x) + E(x)$ 。…… (1)

公式 (1) 中  $E(x)$  称为错误图样。当  $E(x) = 0$  时，则无错误。用  $g(x)$  去除  $R(x)$ ，因为  $C(x)$  是由  $g(x)$  生成的，故  $C(x)$  必能为  $g(x)$  除尽，显然， $R(x)$  与  $E(x)$  为同余式

$(R(x) \equiv E(x) \bmod g(x))$ ，以 $g(x)$ 除 $E(x)$ 所得余式称之为伴随式 $S(x)$ 。

由公式（1）可知， $R(x) \cdot \mathbf{H}^T = (C(x) + E(x)) \cdot H(x) = E(x) \cdot H(x)$ 。若 $E(x) = 0$ ，则 $E(x) \cdot H(x) = 0$ ；若 $E(x) \neq 0$ ，则 $H(x) \neq 0$ 。这说明， $R(x) \cdot \mathbf{H}^T$  仅与错误图样有关，而与发送的码字无关，由此可以确定错误图样表。

由于 $g(x)$ 的次数为 $n - k$ 次， $g(x)$ 除 $E(x)$ 后得余式（即伴随式）的最高次数为 $n-k-1$ 次，故 $S(x)$ 共有 $2^{n-k}$ 个可能的表达式，每一个表达式对应一个错误格式。可以知道（7，4）循环码的 $S(x)$ 共有 $2^{(7-4)} = 8$ 个可能的表达式，可根据错误图样表来纠正（7，4）循环码中的一位错误，其伴随式如表1所示。

表 1 BCH（7，4）循环码错误图样表

错误图样	错误图样码字	伴随式 $S(x)$	伴随式
$E_6(x) = x^6$	1000000	$x^2$	100
$E_5(x) = x^5$	0100000	$x^2 + x$	110
$E_4(x) = x^4$	0010000	$x^2 + x + 1$	111
$E_3(x) = x^3$	0001000	$x + 1$	011
$E_2(x) = x^2$	0000100	$x^2 + 1$	101
$E_1(x) = x^1$	0000010	$x$	010
$E_0(x) = x^0$	0000001	1	001
$E(x) = 0$	0000000	0	000

上式指出了系统循环码的译码方法：将收到的码字 $R(x)$ 用 $g(x)$ 去除，如果除尽则无错；否则有错。如果有错，可由余式 $S(x)$ 一一找出对应图样，然后将错误图样 $E(x)$ 与 $R(x)$ 模2和，即为所求码字 $C(x)$ ，从而实现纠错目的。

需要指出的是，有错码的接收码组也有可能被 $g(x)$ 整除，这时的错码就不能检出了。这种错误被称为不可检错误，不可检错误中的错码数必将超过这种编码的检错能力。

在接收端为纠错而采用的译码方法自然比检错要复杂许多，因此，对纠错码的研究大都集中在译码算法上。

我们知道，校正子与错误图样之间存在某种对应关系。如同其它线性分组码，循环编码和译码可以分三步进行：

- （1）由接收到的码多项式 $B(x)$ 计算校正子（伴随式）多项式 $S(x)$ ；

(2) 由校正子  $S(x)$  确定错误图样  $E(x)$ ;

(3) 将错误图样  $E(x)$  与  $B(x)$  相加, 纠正错误。

上述第 (1) 步运算和检错译码类似, 也就是求解  $B(x)$  整除  $g(x)$  的余式, 第 (3) 步也很简单。因此, 纠错码译码器的复杂性主要取决于译码过程的第 (2) 步。



## 第4章 卷积码的理论及编码译码方式

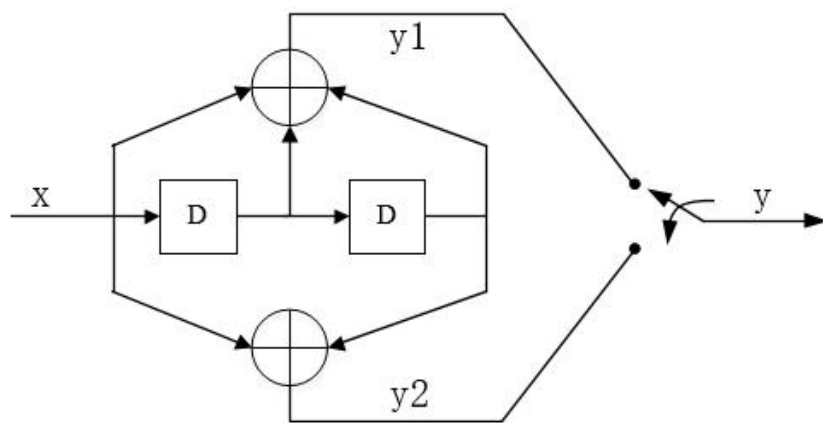
### 4.1 引言

卷积码是非分组码的一个例子。在卷积码中，并不是使用一组信息符号来计算奇偶校验符号，而是利用一段信息符号来计算。在这里，“段”指的是每次当一个信息符号输入编码器，所发生的状态变换而产生的符号。

### 4.2 编码原理

#### 4.2.1 编码器结构

卷积码编码器的核心是存储 $L$ 组信息符号的移位寄存器。每当有一个新的元素进入编码器时，寄存器会将最早进入的元素丢弃，然后将元素整体移位，再将最新进入的元素存储，在这个过程中，会产生一串新的输出编码串。



编码器的整体结构如上图所示，在这里用一个具体的编码器来进行说明。 $x$ 为输入的比特（组）， $D$ 为移存器， $y_1, y_2$ 为每次输出的码元，通过一个电子开关连接至输出 $y$ ，每当完成一次输入的时候，电子开关会完成一次旋转，依次将输出码元输出。在这个例子中，每次输入1比特码元，输出2比特码元。可以看见， $y_1, y_2$ 的生成是遵从一定规律的，都是将某几个寄存器的值进行模2的加法后生成。可以

看见，输出码元的生成法则有很多种可能性，不同的法则也会对编码效果产生较大的影响，具体如何连接是需要进行设计的。

在描述卷积码编码器的时候，我们常使用 3 个参数， $(n, k, L)$ ，这三个参数也对应着编码器的一定结构。 $n$ 表示每一次输出会输出 $n$ 个码元， $k$ 表示每一组输入需要输入 $k$ 个码元， $L$ 表示记忆深度，也就是编码器对之前输入记忆的能力，在这个实例中，可以记住前 2 次输入，因此 $L = 2$ 。在上图这个实例中，我们可以称之为 $(2, 1, 2)$ 编码器。

但是仅仅使用上述 3 个参数，还是不足以完全描述编码器结构的，我们还是需要对输出的连接结构进行描述才能够确定编码器的完整结构。还是以上图为例，我们将输入，以及两个移存器的输出依次描述为 $S_1, S_2, S_3$ ，那么我们可以看见，这个编码器的输出有以下关系

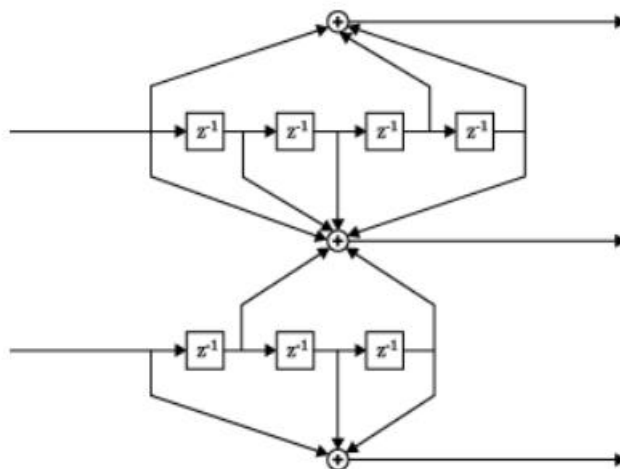
$$y_1 = S_1 \oplus S_2 \oplus S_3$$

$$y_2 = S_1 \oplus S_3$$

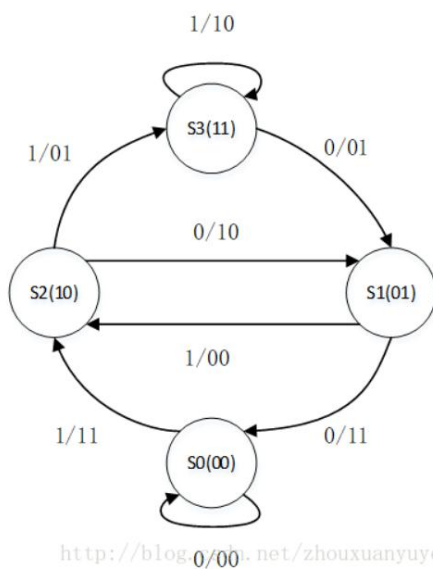
有了这两个条件，我们就能比较完整地描述一个编码器的结构了。事实上，我们可以使用其他方式来描述，例如，生成矩阵

$$G = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

在 MATLAB 中，编码器结构的生成使用 `poly2trellis(ConstraintLength, CodeGenerator)` 函数，这个函数使用两个矩阵参数，其中第一个参数描述了每个输入所对应的记忆深度+1，在此例中为 $[2]$ ，第二个参数描述了每个输出与各状态的连接关系，其中每个元素使用 8 进制来替代二进制表示，其中的元素 $(i, j)$ 描述了第 $i$ 个输入对于第 $j$ 个输出的影响。具体地说，分析上图结构，仅有 1 个输入，因此仅有 1 行元素；有 2 个输出，因此有 2 列元素，这是个 $1 \times 2$ 的矩阵，第一个元素我们可以看见， $y_1$ 的连接方式可以用二进制表示为 111，转换为八进制即为 7，同理第二个元素使用八进制表示应为 5，因此这个函数的第二个参数在此例中应当为 $[7 \ 5]$ 。



再使用一个实例，如上图所示。其有 2 个输入，3 个输出，从上往下分别标号为 1, 2 以及 1, 2, 3。第 1 个输入的记忆深度为 4，第 2 个输入的记忆深度为 3，因此有  $\text{ConstraintLength} = [5 \ 4]$ 。第 1 个输入对第 1 个输出的贡献用二进制表示为 10011，转换为八进制即为 23，同理可以表示另外 5 个元素，最终可以得到  $\text{CodeGenerator} = [23 \ 35 \ 0; \ 0 \ 5 \ 13]$ 。



#### 4.2.2 状态图

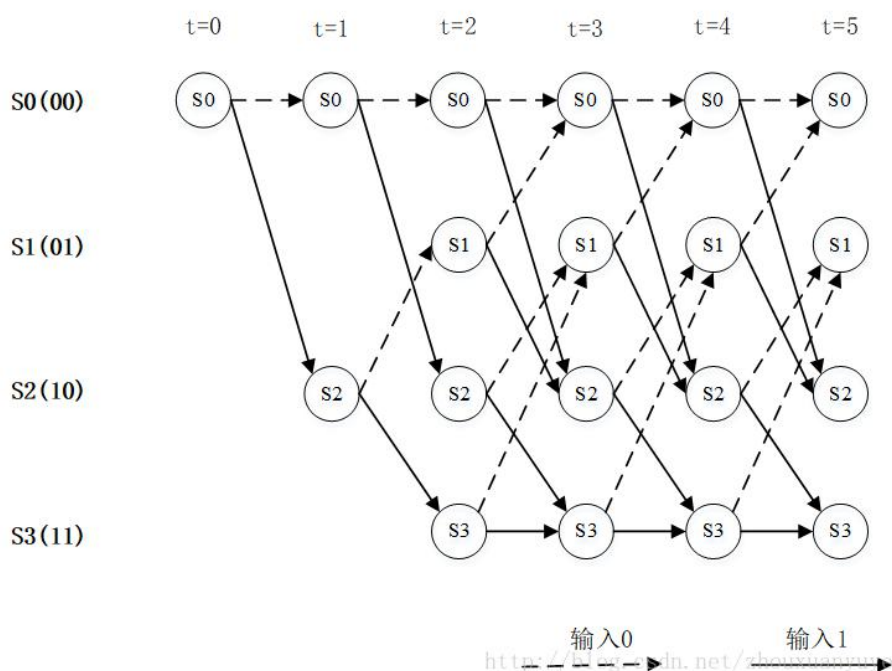
在对编码器结构有了了解之后，具体的编码输出就可以比较轻松地获取了。为

为了更好地描述编码器，我们引入状态图。可以发现，一个记忆深度为 $L$ 的编码器有 $2^L$ 种状态，以(2, 1, 2)编码器为例，两个寄存器的状态可以有00, 01, 10, 11共四种状态。而输入有可能为0或1，对于每次输入，寄存器会抛弃最早进入的数据，然后移位，存储最新的输入，也就是说每次输入都会改变编码器的状态。同时，在新输入进入的时候，会有一个输出，这个输出与编码器的状态以及输入值都有关系。将这些信息整合到一张图上，就可以得到编码器的状态图。(2, 1, 2)编码器的状态图如上图所示。

在图中， $S_n(ij)$ 表示当寄存器状态为 $ij$ 时对应第 $n$ 个状态，每条状态转移线上 $x/mn$ 表示在输入 $x$ 时，寄存器的状态由起点转移至终点，同时输出 $mn$ 。有了状态转移图，当任意给定输入序列时，可以非常方便地查找出状态的转移以及编码器的输出序列。

### 4.2.3 篱笆图（网格图）

但是仅仅只有状态转移图仍然不足以描述随着时间的推进（或者说序列的输入），编码器输出以及状态的可能路径，因此引入了篱笆图。上述(2,1,2)编码器的篱笆图如下图所示。



图中实线箭头表示在某一状态下，输入 1 后状态转移的路径；虚线箭头表示在某一状态下，输入 0 后状态转移的路径。

有了篱笆图，我们就可以看到较为完整的，当输入某一序列串后，编码器状态转移的路径，以及输出序列（这张图中并没有在每条路径上标出输出，但通常我们会将输出也标出来）。

另外，我们可以看到，在  $t = 2$  时刻之后，每一次输入编码器结构的跳变组合都是一致的，篱笆图的结构开始重复出现，我们称之为编码器各个稳定状态的状态转移。

在 MATLAB 的 `trellis` 结构中，也有着对应的描述矩阵。`nextStates` 矩阵即为状态转移矩阵，在上述(2,1,2)编码器对应的 `trellis` 结构中，有

$$\text{nextStates} = \begin{bmatrix} 0 & 2 \\ 0 & 2 \\ 1 & 3 \\ 1 & 3 \end{bmatrix}$$

其中  $(i, j)$  位置对应的元素的含义为：当编码器处于状态  $i$  的时候（ $i = 1$  对应 00,  $i = 4$  对应 11），输入  $j$  号元素（ $j = 1$  对应输入 0,  $j = 2$  对应输入 1）的时候，跳转至这个位置对应元素所表示的状态，这些状态以十进制表示。

而 `trellis` 结构中 `output` 矩阵为输出矩阵，在上述(2,1,2)编码器对应的 `trellis` 结构中，有

$$\text{output} = \begin{bmatrix} 0 & 3 \\ 3 & 0 \\ 2 & 1 \\ 1 & 2 \end{bmatrix}$$

其中  $(i, j)$  位置对应元素的含义为：当编码器处于状态  $i$  的时候（ $i = 1$  对应 00,  $i = 4$  对应 11），输入  $j$  号元素（ $j = 1$  对应输入 0,  $j = 2$  对应输入 1）的时候，编码器输出的编码，这些输出以八进制表示。对于结构更加复杂的编码器，`trellis` 结构内容同理。

### 4.3 维特比（Viterbi）算法译码

卷积码解码的基本思路为从篱笆图中所有可能的路径中，找到最有可能是原信

息编码时走过的那条路径。理论上讲，我们可以遍历每一条路径，然后计算找出每条路径所输出的编码与需要解码的编码串最相似的一条路径，从而实现解码。但是当原信息有 $l$ 位时，可能的路径有 $2^l$ 条，随着原信息的长度增长，计算量呈指数增长，这在实际上是基本不可能实现的，对算力的要求也是及其巨大的。

维特比算法在编码器达到稳定状态后，每次删除一半累积汉明距离较大的路径，从而将指数级增长的计算量转换成了线性增长的计算量，对算力的要求极大减小了。维特比算法被广泛应用于通信系统中，可以证明，维特比算法属于最大似然译码，在这个意义上维特比算法为最佳译码。此处以上述(2,1,2)编码器的译码为例来说明这个算法的流程。

我们将一串比特 10010 进行编码，由状态转移图，从状态 00 开始，可以得到编码的结果 1110111110。在信道噪声的干扰下，编码的某一位出现了错误，假设接收机解调结果为 1110111010。

接下来对其进行译码，译码算法的流程基于前述篱笆图上。算法的第一步，由于编码器初始状态为 00，因此从 $S_1$ 开始，需要经过 2 步才有能达到编码器的稳定状态（因为有 2 个初始为 0 的寄存器），因此我们要在解码的时候，最开始需要先走 2 步，得到 4 条可能的路径，然后比较这 4 条可能路径再经过一步后累积的汉明距离。也就是说我们需要计算输入编码的前 $2 \times 3 = 6$ 位与前 3 步可能输出的汉明距离。 $a$ 和 $b$ 的汉明距离即为将两者逐位异或后所得结果中 1 的个数。结果列于下表之中。

前两步经历的状态	总经历的状态	对应的输出编码	汉明距离	是否幸存
$S_0S_0S_0$	$S_0S_0S_0S_0$	000000	5	否
	$S_0S_0S_0S_2$	000011	3	是
$S_0S_0S_2$	$S_0S_0S_2S_1$	001110	4	是
	$S_0S_0S_2S_3$	001101	4	否
$S_0S_2S_1$	$S_0S_2S_1S_0$	111011	0	是
	$S_0S_2S_1S_2$	111000	2	否
$S_0S_2S_3$	$S_0S_2S_3S_1$	110101	3	是

	$S_0S_2S_3S_3$	110110	3	否
--	----------------	--------	---	---

第一步的第二部分，是将 4 组中每组内两条路径累积的汉明距离进行比较，然后删除汉明距离较大的那一条，剩下的一条路径称之为幸存路径。当同组内两条路径累积的汉明距离相同的时候，任意留下一条而删去另一条。最后幸存判断结果列于上表。

接下来进行第二步，由于编码器已经达到了稳定状态，因此每前进一步都重复一遍计算累积汉明距离，留下幸存路径的过程。注意在计算汉明距离的时候，应当将之前的所有汉明距离都累积起来，两次计算过程也分别列于下方。

之前经历的状态	最新的状态	最后一步的输出	累积汉明距离	是否幸存
$S_0S_0S_0S_2$	$S_1$	10	$3 + 0 = 3$	是
	$S_3$	01	$3 + 2 = 5$	否
$S_0S_0S_2S_1$	$S_0$	11	$4 + 1 = 5$	是
	$S_2$	00	$4 + 1 = 5$	否
$S_0S_2S_1S_0$	$S_0$	00	$0 + 1 = 1$	否（是）
	$S_2$	11	$0 + 1 = 1$	是（否）
$S_0S_2S_3S_1$	$S_0$	11	$3 + 1 = 4$	是
	$S_2$	00	$3 + 1 = 4$	否

之前经历的状态	最新的状态	最后一步的输出	累积汉明距离
$S_0S_0S_0S_2S_1$	$S_0$	11	$3 + 1 = 4$
	$S_2$	00	$3 + 1 = 4$
$S_0S_0S_2S_1S_0$	$S_0$	00	$5 + 1 = 6$
	$S_2$	11	$5 + 1 = 6$
$S_0S_2S_1S_0S_2$	$S_1$	10	$1 + 0 = 1$
	$S_3$	01	$1 + 2 = 3$
$S_0S_2S_3S_1S_0$	$S_0$	00	$4 + 1 = 5$
	$S_2$	11	$4 + 1 = 5$

当到达最后一组编码，比较剩余所有路径的累积汉明距离，取最短的一条路径，也就是 $S_0S_2S_1S_0S_2S_1$ ，其对应的解码结果为 10010，可以发现与原信息一致，也就是



说维特比算法确实可以纠错，从而一定程度上抗噪声影响。

第三步，为了解码下一组输入，我们需要将整个篱笆图状态归 0，由于编码器有 2 个寄存器，因此我们要虚拟地在原信息末尾插入 2 个 0，从而使编码器归 0。

事实上，在上述解码过程中，第二次选择幸存路径的时候很幸运地选中了正确的路径，接下来演示一下如果 $s_0s_2s_1s_0$ 后一个状态选择了 $s_0$ 作为幸存路径将出现的结果。

之前经历的状态	最新的状态	最后一步的输出	累积汉明距离
$s_0s_0s_0s_2s_1$	$s_0$	11	$3 + 1 = 4$
	$s_2$	00	$3 + 1 = 4$
$s_0s_0s_2s_1s_0$	$s_0$	00	$5 + 1 = 6$
	$s_2$	11	$5 + 1 = 6$
$s_0s_2s_1s_0s_0$	$s_0$	00	$1 + 1 = 2$
	$s_2$	11	$1 + 1 = 2$
$s_0s_2s_3s_1s_0$	$s_0$	00	$4 + 1 = 5$
	$s_2$	11	$4 + 1 = 5$

可以看见累积汉明距离最小的有两条路径，分别对应的解码结果为 10000 以及 10001，也就是说，在有噪声影响，出现传输错误的时候，即使只有 1 位出现了错误，卷积码也并不能保证能够完全纠错，这与汉明码是不同的。在这种情况下，可以看见编码器经历的状态已经偏离正确路径了，误码率也可能为 0%，20%，甚至 40%。但是如此高的误码率是由于原信息长度过短，当信息足够长的时候，编码器早已进入稳定状态，每次的状态转移是相同的，根据马尔可夫过程理论，这是一个具有遍历性的齐次马尔科夫链，当时间趋近于无穷时，出现在各个状态的概率应当是相等的，所以在经历一段错误之后，能够再次回到正确的路径上。

从这个实例中，我们也可以看出，卷积码编码后的信息，前后是有相互关联的，因此当一位解码出错，有可能导致一串解码出错，因此在低信噪比的情况下可能会出现误码率比未编码情况下还要高的情况，这一点在之后的仿真中也有体现。



## 第 5 章 仿真结果与理论分析

仿真整体流程图如下图 5.0 所示：

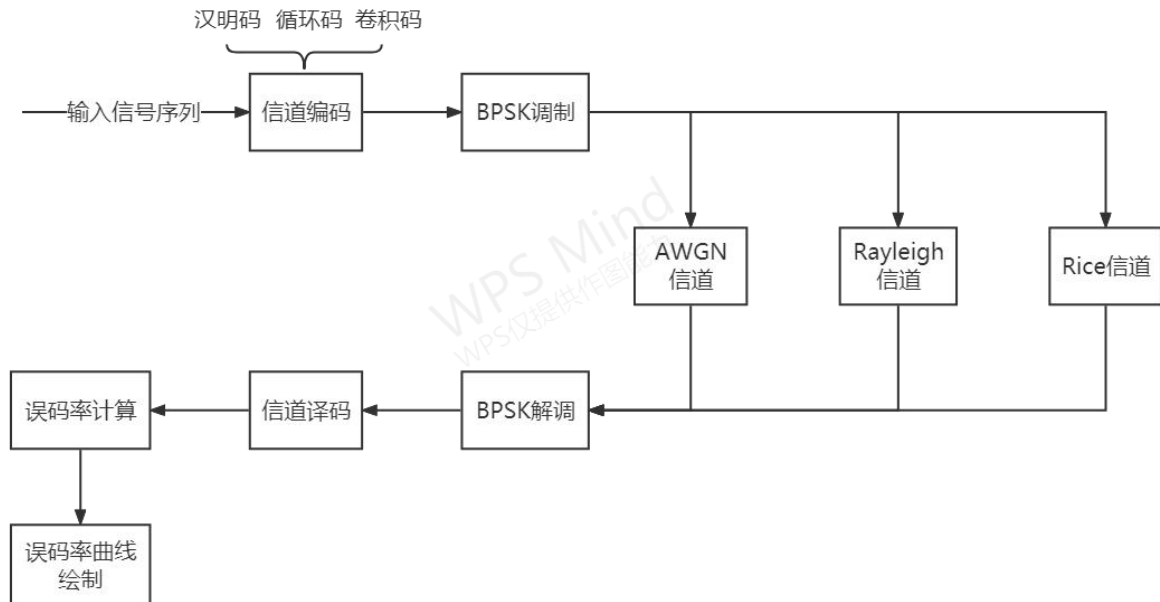


图5.0

## 5.1 对于同一编码方式不同信道之间的误码率曲线关系

### 5.1.1 对于汉明码编码方式不同信道之间的误码率曲线关系

进行汉明码编码后经过不同信道之间的误码率曲线关系如下图 5.1.1 所示

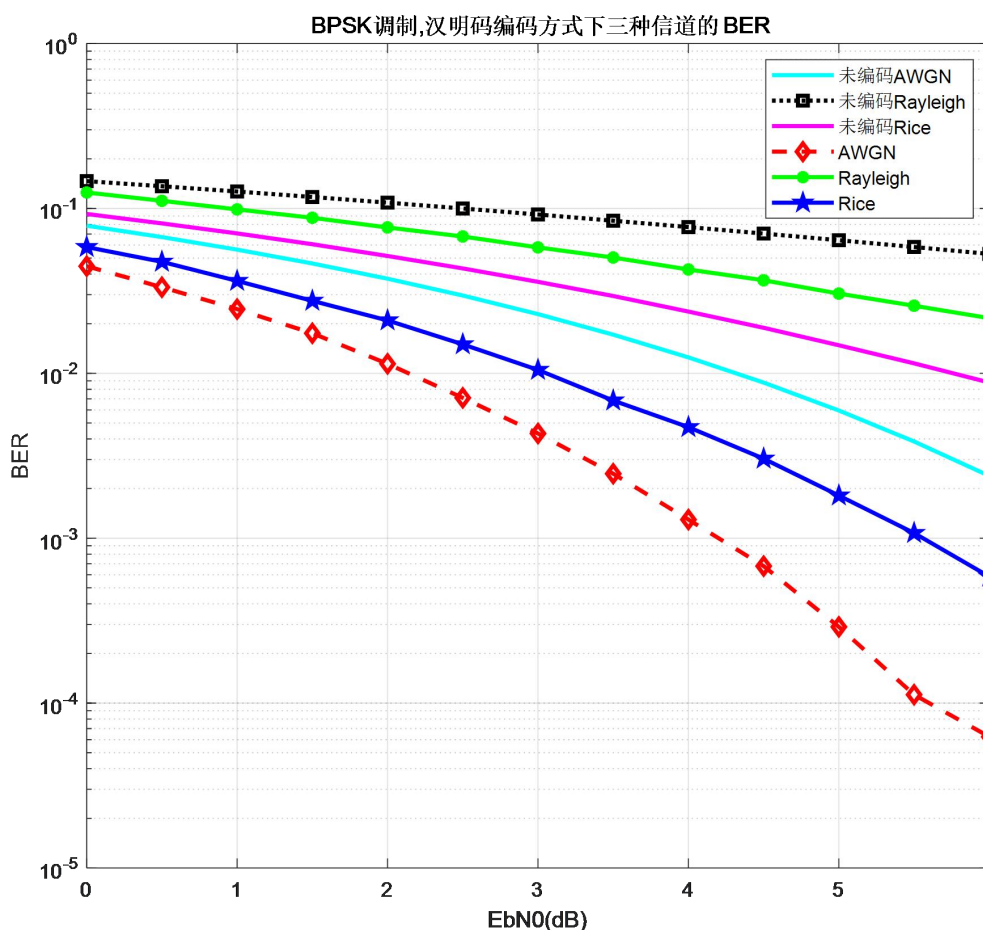


图 5.1.1

仿真结果分析：首先对于未编码的信号经过三种不同信道之后的误码率进行分析，从图 5.1.1 中可以看出未编码信号通过三种不同信道后误码率由小到大依次是 AWGN 信道、瑞利信道、莱斯信道，说明对于信号的影响由小到大的信道依次是 AWGN 信道、瑞利信道、莱斯信道；其次对于经过汉明码编码后的信号经过三种不同信道之后的误码率进行分析，从图 5.1.1 中可以看出经过汉明码编码后的信号通过三种不同信道后误码率由小到大的排序仍为 AWGN 信道、瑞利信道、莱斯信道，与

未编码得到的结果一致，但是对于整体的误码率进行对比可发现，经过汉明码编码后的信号经过三种不同信道后整体误码率都比未编码时低，且差距较为明显，在高信噪比下可以产生至少两个数量级的差距，说明汉明码编码对于信号的正确编译性能较为良好。

### 5.1.2 对于循环码编码方式不同信道之间的误码率曲线关系

进行循环码编码后经过不同信道之间的误码率曲线关系如下图 5.1.2 所示

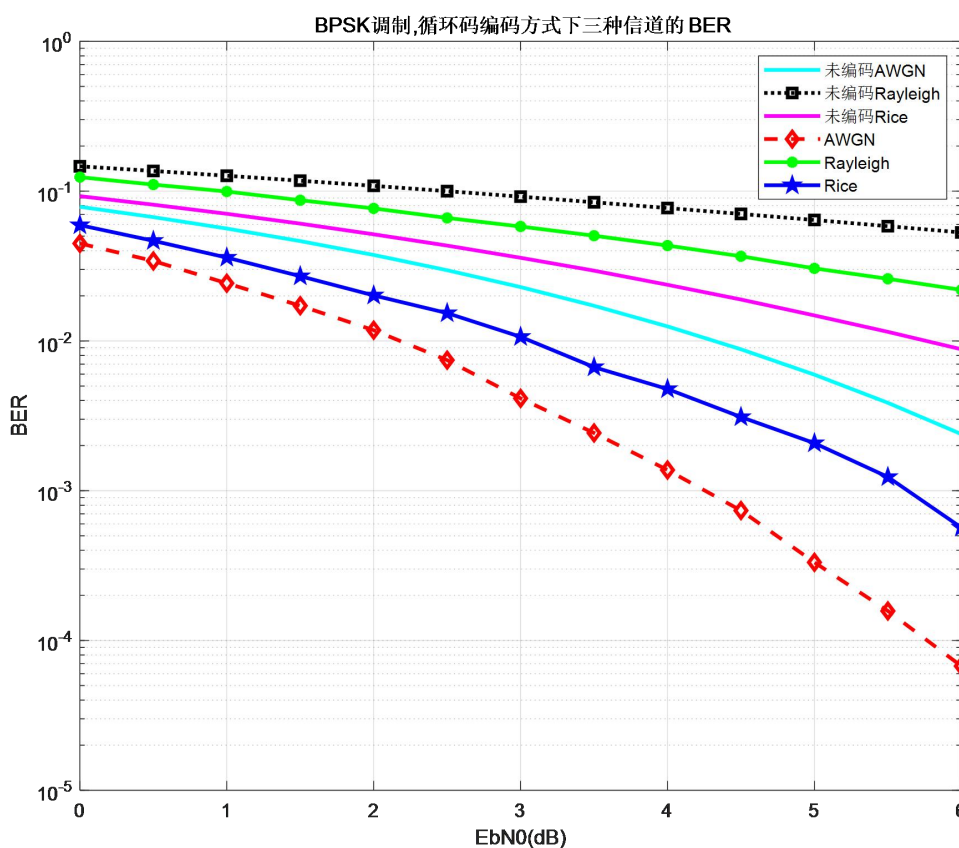


图 5.1.2

仿真结果分析：首先对于未编码的信号经过三种不同信道之后的误码率进行分析，从图 5.1.2 中可以看出未编码信号通过三种不同信道后误码率由小到大依次是 AWGN 信道、瑞利信道、莱斯信道，说明对于信号的影响由小到大的信道依次是 AWGN 信道、瑞利信道、莱斯信道；其次对于经过循环码编码后的信号经过三种不同信道之后的误码率进行分析，从图 5.1.2 中可以看出经过循环码编码后的信号

通过三种不同信道后误码率由小到大的排序仍为 AWGN 信道、瑞利信道、莱斯信道，与未编码得到的结果一致，但是对于整体的误码率进行对比可发现，经过循环码编码后的信号经过三种不同信道后整体误码率都比未编码时低，且差距较为明显，在高信噪比下可以产生至少两个数量级的差距，说明循环码编码对于信号的正确编译性能较为良好。进一步与图 5.1.1 汉明码的编码过程进行横向对比可获得更进一步的结论，二者的误码率相差无几，说明二者的编码方式产生的效果与性能无较大差距，从理论分析可知，二者均属于线性分组码，所以在译码后得到的误码率大小并无较大差距。

### 5.1.3 对于卷积码编码方式不同信道之间的误码率曲线关系

#### 5.1.3.1 对于(3,1,2)卷积码编码方式不同信道之间的误码率曲线关系

进行 (3, 1, 2) 卷积码编码后经过不同信道之间的误码率曲线关系如下图

5.1.3.1 所示

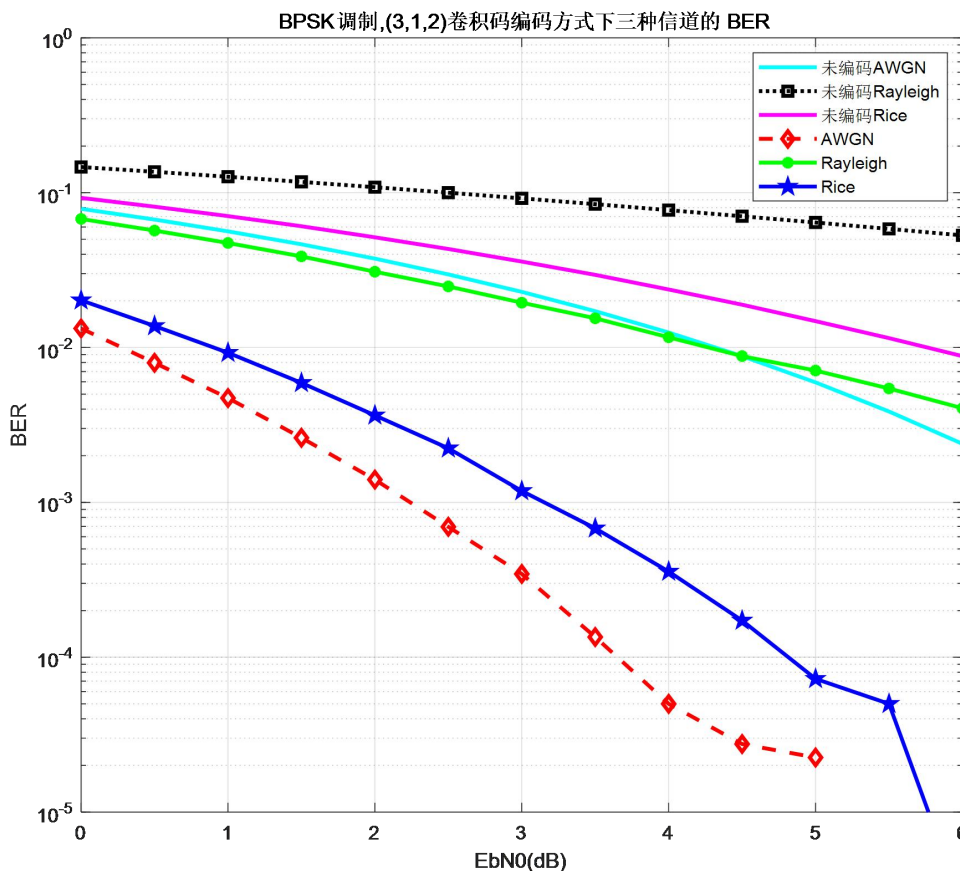


图 5.1.3.1

仿真结果分析：首先对于未编码的信号经过三种不同信道之后的误码率进行分析，从图 5.1.3.1 中可以看出未编码信号通过三种不同信道后误码率由小到大依次是 AWGN 信道、瑞利信道、莱斯信道，说明对于信号的影响由小到大的信道依次是 AWGN 信道、瑞利信道、莱斯信道；其次对于经过(3,1,2)卷积码编码后的信号经过三种不同信道之后的误码率进行分析，从图 5.1.3.1 中可以看出经过(3,1,2)卷积码编码后的信号通过三种不同信道后误码率由小到大的排序仍为 AWGN 信道、瑞利信道、莱斯信道，与未编码得到的结果一致，但是对于整体的误码率进行对比可发现，经过(3,1,2)卷积码编码后的信号经过三种不同信道后整体误码率都比未编码时低，且差距较为明显，在高信噪比下可以产生至少三个数量级的差距，说明(3,1,2)卷积码编码对于信号的正确编译性能较为良好，与上述汉明码和循环码进一步进行横向对比可得，在高信噪比下，(3,1,2)卷积码的性能要优于汉明码和循环码

的性能，译码后信号的误码率也比汉明码和循环码的误码率低。

### 5.1.3.2 对于(2,1,2)卷积码编码方式不同信道之间的误码率曲线关系

进行 (2, 1, 2) 卷积码编码后经过不同信道之间的误码率曲线关系如下图 5.1.3.1 所示

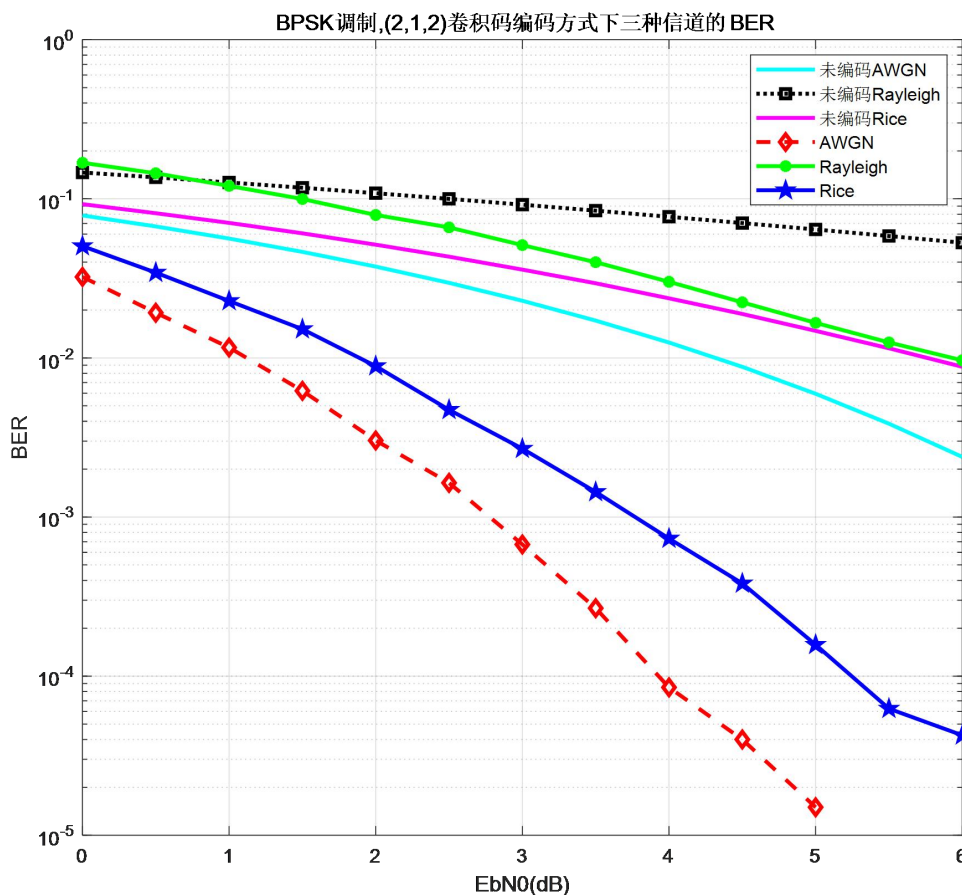


图 5.1.3.2

仿真结果分析：首先对于未编码的信号经过三种不同信道之后的误码率进行分析，从图 5.1.3.2 中可以看出未编码信号通过三种不同信道后误码率由小到大依次是 AWGN 信道、瑞利信道、莱斯信道，说明对于信号的影响由小到大的信道依次是 AWGN 信道、瑞利信道、莱斯信道；其次对于经过(2,1,2)卷积码编码后的信号经过三种不同信道之后的误码率进行分析，从图 5.1.3.2 中可以看出经过(2,1,2)卷积码编码后的信号通过三种不同信道后误码率由小到大的排序仍为 AWGN 信道、瑞

利信道、莱斯信道，与未编码得到的结果一致，但是对于整体的误码率进行对比可发现，经过(2,1,2)卷积码编码后的信号经过三种不同信道后在高信噪比的情况下整体误码率都比未编码时低，且差距较为明显，在高信噪比下可以产生至少三个数量级的差距，说明(2,1,2)卷积码编码对于信号的正确编译性能较为良好，与上述汉明码和循环码进一步进行横向对比可得，在高信噪比下，(2,1,2)卷积码的性能要优于汉明码和循环码的性能，译码后信号的误码率也比汉明码和循环码的误码率低。而在低信噪比时，以瑞利信道为例，发现利用卷积码进行编码后信号的误码率反而比未编码时更大，这是由于在信噪比较低时，卷积码使得状态路径偏离正确路径导致出现一串译码错误，在实际运用中由于卷积码在高信噪比的优越性能，常用于信噪比较高时。

#### 5.1.3.3 对于(2,1,4)卷积码编码方式不同信道之间的误码率曲线关系

进行(2, 1, 4)卷积码编码后经过不同信道之间的误码率曲线关系如下图

##### 5.1.3.1 所示

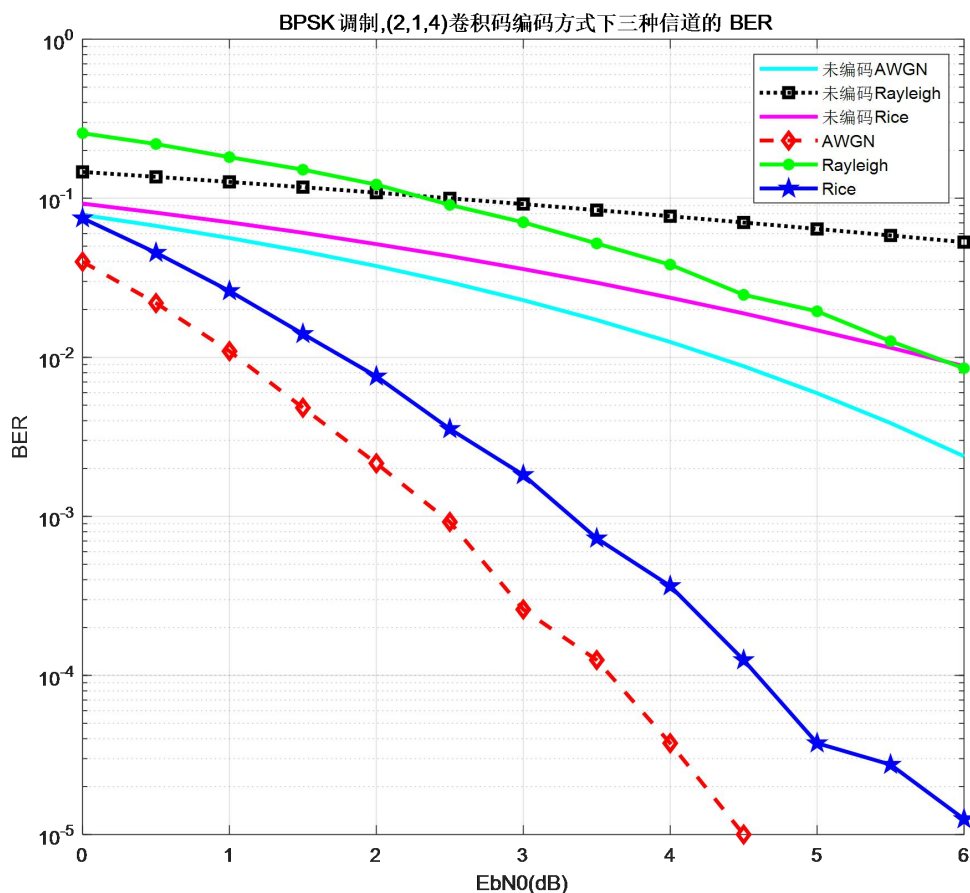


图 5.1.3.3

仿真结果分析：首先对于未编码的信号经过三种不同信道之后的误码率进行分析，从图 5.1.3.3 中可以看出未编码信号通过三种不同信道后误码率由小到大依次是 AWGN 信道、瑞利信道、莱斯信道，说明对于信号的影响由小到大的信道依次是 AWGN 信道、瑞利信道、莱斯信道；其次对于经过(2,1,4)卷积码编码后的信号经过三种不同信道之后的误码率进行分析，从图 5.1.3.3 中可以看出经过(2,1,4)卷积码编码后的信号通过三种不同信道后误码率由小到大的排序仍为 AWGN 信道、瑞利信道、莱斯信道，与未编码得到的结果一致，但是对于整体的误码率进行对比可发现，经过(2,1,4)卷积码编码后的信号经过三种不同信道后在高信噪比的情况下整体误码率都比未编码时低，且差距较为明显，在高信噪比下可以产生至少三个数量级的差距，说明(2,1,4)卷积码编码对于信号的正确编译性能较为良好，与上述



汉明码和循环码进一步进行横向对比可得，在高信噪比下， $(2,1,4)$ 卷积码的性能要优于汉明码和循环码的性能，译码后信号的误码率也比汉明码和循环码的误码率低。而在低信噪比时，以瑞利信道为例，发现利用卷积码进行编码后信号的误码率反而比未编码时更大，这是由于在信噪比较低时，卷积码译码状态路径容易偏离正确路径，导致经常出现一连串译码错误。在实际运用中由于卷积码在高信噪比的优越性能，常用于信噪比较高时。

## 5.2 对于同一信道不同编码方式之间的误码率曲线关系

### 5.2.1 对于 AWGN 信道中不同编码方式的误码率曲线关系

经过 AWGN 信道的不同编码方式误码率曲线关系如下图 5.2.1 所示

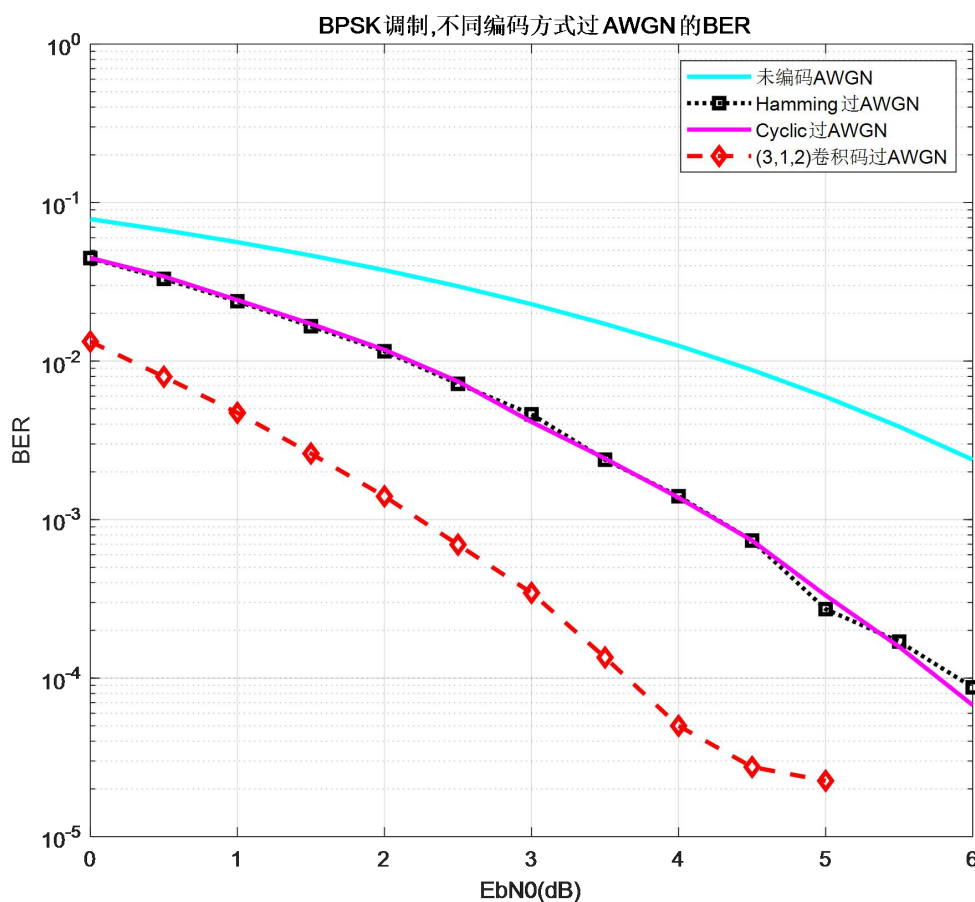


图 5.2.1

仿真结果分析：从图 5.2.1 中可得，经过 AWGN 信道后误码率从小到大排序依次是(3,1,2)卷积码、循环码、汉明码、未编码信号，其中(3,1,2)卷积码的性能要远优于其他两种编码方式，误码率也与其余两种编码方式有一个数量级的差距，而循环码和汉明码的误码率基本一致，说明二者的编码方式产生的效果与性能无较大差距，从理论分析可知，二者均属于线性分组码，所以在译码后得到的误码率大小并无较大差距。

## 5.2.2 对于瑞利信道中不同编码方式的误码率曲线关系

经过瑞利信道的不同编码方式误码率曲线关系如下图 5.2.2 所示。

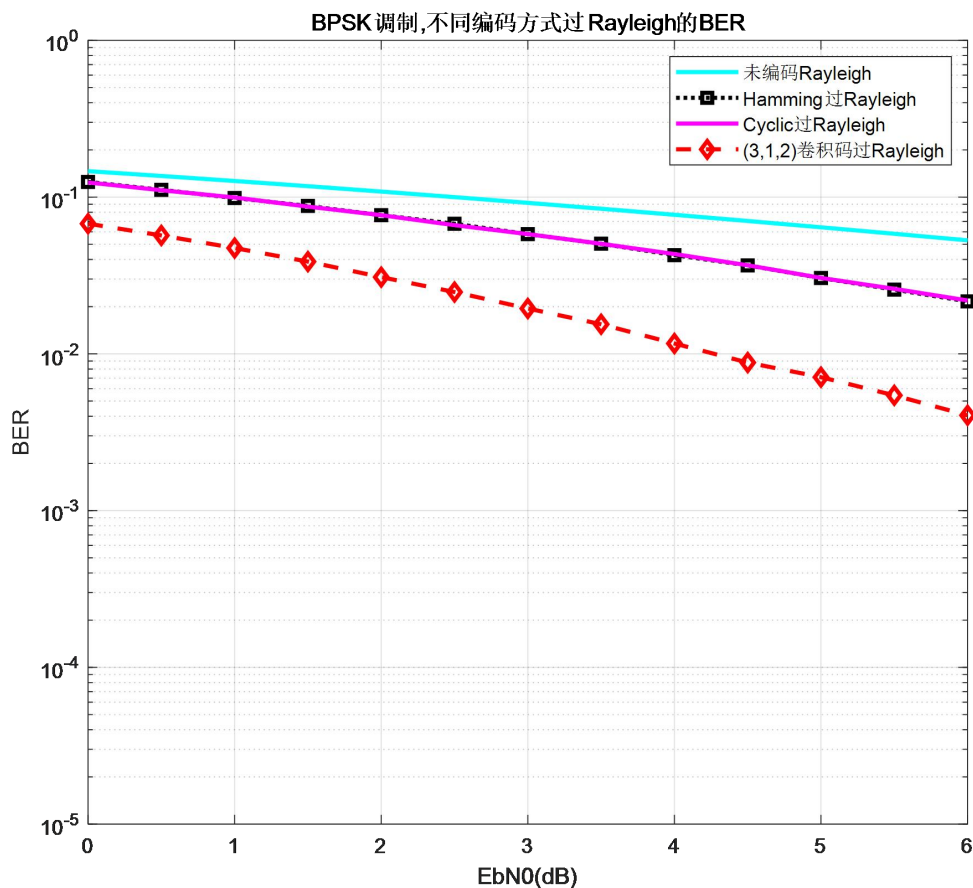


图 5.2.2

仿真结果分析：从图 5.2.2 中可得，经过瑞利信道后误码率从小到大排序依次是(3,1,2)卷积码、循环码、汉明码、未编码信号，其中(3,1,2)卷积码的性能要远优于其他两种编码方式，误码率也与其余两种编码方式有一个数量级的差距，而循环码和汉明码的误码率基本一致，说明二者的编码方式产生的效果与性能无较大差距，从理论分析可知，二者均属于线性分组码，所以在译码后得到的误码率大小并无较大差距。进一步与图 5.2.1 进行比较可知，在瑞利信道中误码率普遍要大于在 AWGN 信道中，也再次验证瑞利信道对于信号的影响要大于 AWGN 信道。

### 5.2.3 对于莱斯信道中不同编码方式的误码率曲线关系

经过莱斯信道的不同编码方式误码率曲线关系如下图 5.2.3 所示

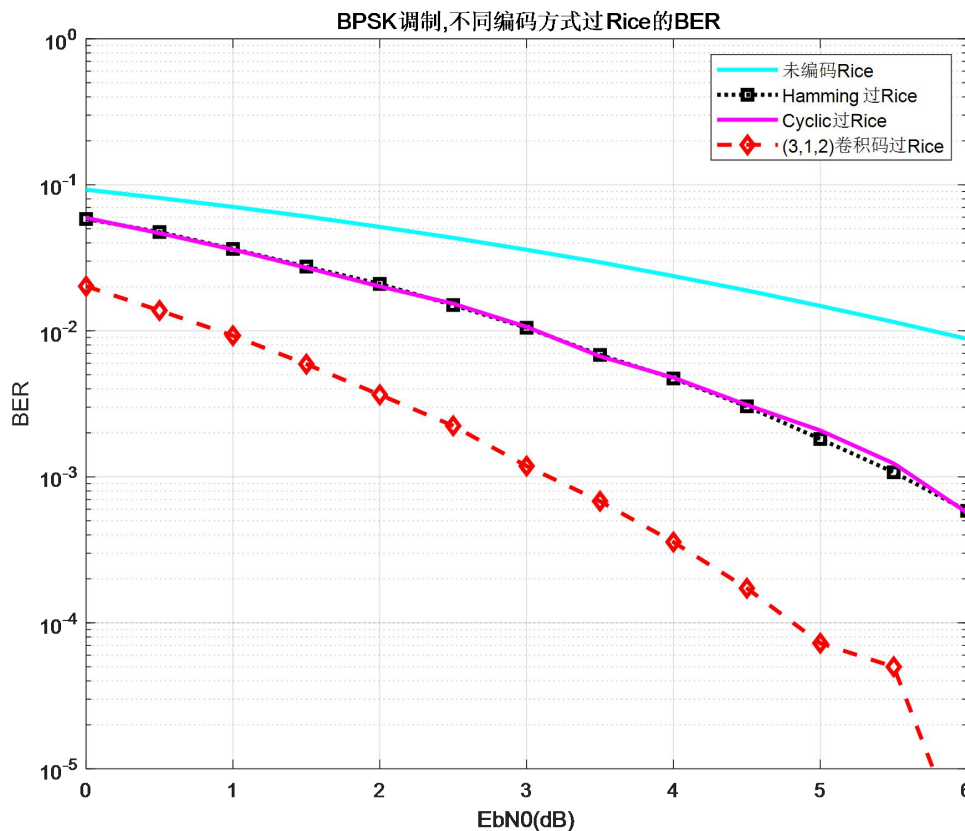


图 5.2.3

仿真结果分析：从图 5.2.3 中可得，经过莱斯信道后误码率从小到大排序依次是(3,1,2)卷积码、循环码、汉明码、未编码信号，其中(3,1,2)卷积码的性能要远优于其他两种编码方式，误码率也与其余两种编码方式有一个数量级的差距，而循环码和汉明码的误码率基本一致，说明二者的编码方式产生的效果与性能无较大差距，从理论分析可知，二者均属于线性分组码，所以在译码后得到的误码率大小并无较大差距。进一步与图 5.2.1 和图 5.2.2 进行比较可知，三种不同信道对于信号的影响由大到小排序依次是瑞利信道、莱斯信道、AWGN 信道。

### 5.3 对于瑞利信道下不同种卷积码之间的误码率曲线关系

进行 (3, 1, 2) 卷积码编码后经过不同信道之间的误码率曲线关系如下图 5.1.3.1 所示

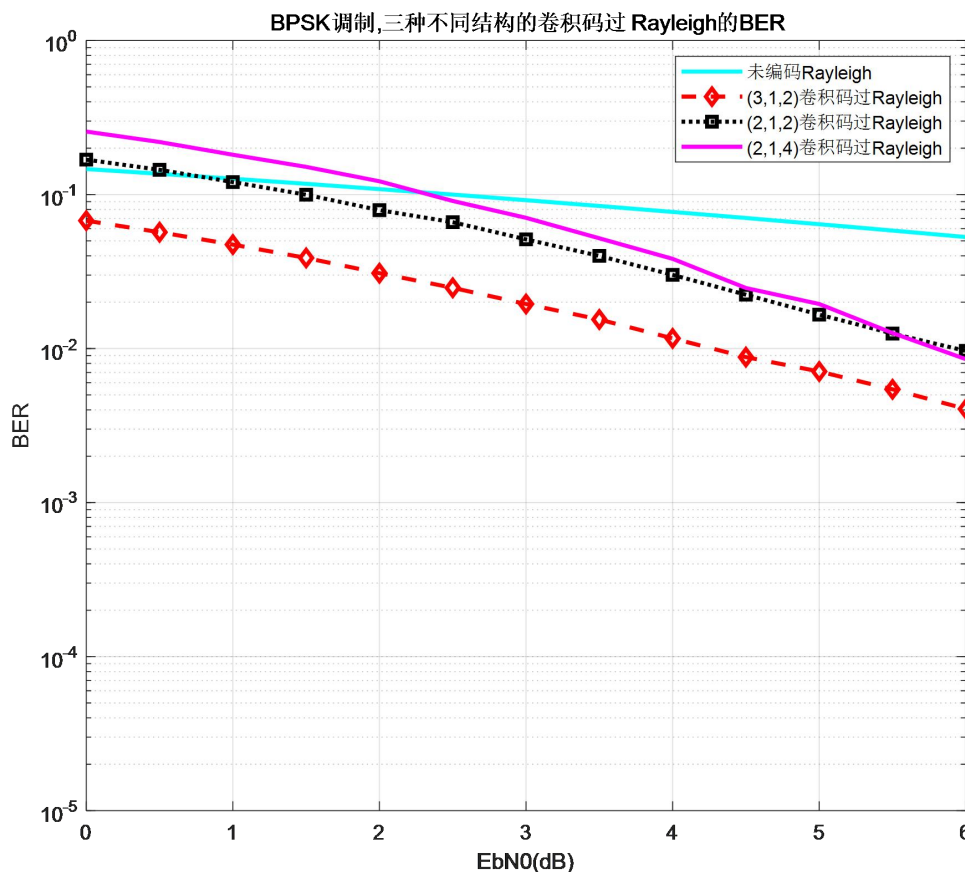


图 5.3

仿真结果分析：首先从理论分析角度明确不同种类卷积码的性能影响因素，(3, 1, 2) 卷积码比 (2, 1, 2) 卷积码和 (2, 1, 4) 卷积码的输出多一位，因此增加的冗余较多，故会使译码后误码率更小；而对于 (2, 1, 2) 卷积码和 (2, 1, 4) 卷积码而言，由于 (2, 1, 4) 卷积码比 (2, 1, 2) 卷积码的编码器内部多出两个寄存器，会导致 (2, 1, 4) 卷积码编码后信号的相关性强于 (2, 1, 2) 卷积码；其次利用理论分析得出的结论来对图 5.3 进行分析，首先 (3, 1, 2) 卷积码由于输出冗余更多，导致误码率会低于其余两种卷积码，与图像体现一致，其次，由于 (2, 1, 4) 卷积码比 (2, 1, 2) 卷积码的编码器内部多出两个寄存器，会导致 (2, 1, 4) 卷积码编码后信号的相关性

强于  $(2, 1, 2)$  卷积码，使得在低信噪比情况下， $(2, 1, 4)$  卷积码译码时状态路径偏离正确路径后更加难以回到正确路径上，一但出现译码错误，连带出现的错误更多，故在低信噪比的情况下， $(2, 1, 4)$  卷积码比  $(2, 1, 2)$  卷积码的误码率要大。但随着信噪比的增加， $(2, 1, 4)$  卷积码误码率减小的变化趋势比  $(2, 1, 2)$  卷积码更快。

## 结 论

（1）针对三种不同信道，对信号的影响由小到大依次顺序是：瑞利信道、莱斯信道、AWGN 信道，引起误码率由大到小依次顺序是：瑞利信道、莱斯信道、AWGN 信道。

（2）针对三种不同信道编码方式，在较高信噪比下，信号的误码率由大到小的依次顺序是：汉明码、循环码、卷积码，其中汉明码和循环码差距较小，卷积码与二者差距较大。

（3）不同种类的卷积码对于信号的误码率影响不同，在较高信噪比下，卷积码的编码器输出位数越多、编码器内部寄存器越多，信号的误码率越小。

（4）卷积码针对低信噪比时对信号的误码率减小的表现较差，针对高信噪比时减小误码率表现较为突出，因此在实际生活中，利用卷积码作为信道编码常用于信噪比较高的场合。

## 参考文献

- [1] 曾凡鑫. 现代编码技术. 西安: 西安电子科技大学出版社, 2007; 33-42
- [2] 樊昌信. 通信原理. 北京市: 国防工业出版社, 2009; 79-83
- [3] 李鹏. 计算机通信技术及其程序设计. 西安: 西安电子科技大学出版社, 1999.6 ; 67-73
- [4] 徐明远, 邵玉斌 MATLAB 仿真在通信与电子工程中的应用. 西安: 西安电子科技大学出版社, 2005.6; 126
- [5] 曹志刚等著, 现代通信原理, 北京: 清华大学出版社, 2001.5: 65-78
- [6] 吴伟陵等著, 移动通信原理, 北京: 电子工业出版社, 2005: 30-57
- [7] 李建新, 现代通信系统分析与仿真-MATLAB 通信工具箱, 西安: 西安电子科技大学出版社, 2000: 13-45
- [8] 唐向宏, 岳恒立, 郑雪峰 MATLAB 及在电子信息类课程中的应用(第二版). 北京: 电子工业出版社, 2009.6; 240-24