

# 实验八 级联码的仿真实现

## 一、 实验目的

1. 掌握级联码的基本原理与编译码算法；
2. 学习使用 MATLAB 软件实现不同交织结构的级联码交织过程；
3. 学习使用 MATLAB 软件绘制级联码经 AWGN 信道传输后的误码率曲线，并对其进行性能分析。

## 二、 实验原理

### 1. 级联码的基本原理

信道编码定理指出，随着码长的增加，译码错误概率按指数接近于零。因此，为了使编码的纠错能力更为有效，必须使用长码；但随着码长的增加，在一个码组中要求纠正的错误码元数目相应增加，使得译码器的复杂度和计算量也相应增加以至于难以实现。为解决编码性能与设备复杂度之间的矛盾，提出了级联编码的概念，即通过两次或多次编码组合的形式来获得长码。

对于 AWGN 信道，多采用汉明码、BCH 码和卷积码等适合纠正随机独立差错的信道编码方式；对于纯衰落信道，可以采用 RS 码、可纠正突发差错的分组码和卷积码等信道编码方式；但在实际信道中，发生的差错往往是突发与随机共存的混合型差错，级联码就是常见的用于纠正这类混合型差错的信道编码方式。

根据结构不同，级联码可分为串行级联码与并行级联码两类。

最简单的串行级联码由内码和外码组成，其编译码器的结构如图 1 所示。一般利用内码纠正大部分的信道差错，剩余错误靠纠错能力稍弱的外码来克服，内码与外码通过交织连接。

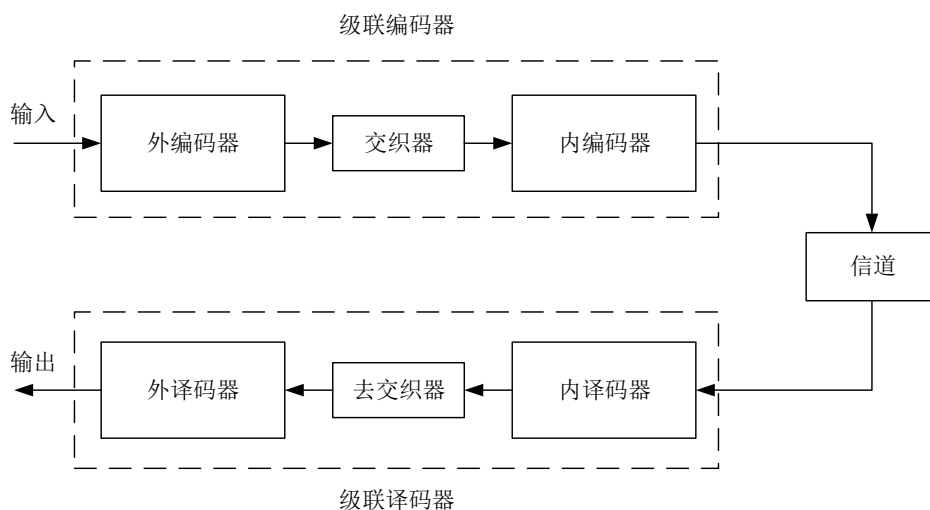


图 1 串行级联编译码器结构框图

串行级联码的内码通常是较短的二进制码，如 BCH 码、卷积码等，而外码可以选择较长的多进制码，如 RS 码。常见的内码选择是卷积码，原因在于卷积码可以进行最优的 Viterbi 译码，同时还可以采用软判决方式以增加编码增益；但卷积码的纠错能力主要依靠相邻码元之间的相关性实现，相关性的维持时间较短，且在低信噪比的情况下，卷积码经 Viterbi 译码后常常产生突发错误。因此，内外码之间的交织器用于离散较长的突发错误，再选择有一定纠正突发错误能力的 RS 码作为外码。

一般来说，在相同的错误概率下，级联编码的复杂度低于单一编码。级联译码的次优方案是顺序地对内码和外码分别进行译码；最优方案是将内码和外码一起进行最大似然译码，但该方案复杂度太高；20 年代 90 世纪中期出现了一种迭代译码方案，能够使级联码的译码性能接近最优，这就是并行级联码，也就是 turbo 码。

## 2. 交织

信道交织编码实际上是一类信道改造技术。发送端将信息序列通过交织器重新排列，接收端将接收序列通过去交织器恢复原序，以求将信道中较长的突发差错离散成随机独立差错。交织编码本身并不具备检纠错的能力，仅起到信号预处理作用，交织前相邻的码元在交织后的最小距离称为交织深度，交织深度越大，抗突发干扰能力越强，但带来的时延也越大。交织深度应大于信道突发错误的长度。

常见的交织结构包括分组交织、卷积交织与随机交织。

### 分组交织：

$(m, n)$  分组交织器的交织过程如图 2 所示：将信息序列按行写入  $m \times n$  矩阵，然后按列读出；交织深度为行数  $m$ ；交织器带来的固定时延为  $mn$  个码元周期，硬件实现上需占据  $2 \times m \times n$  个移位寄存器。

相应去交织器的去交织过程就是将交织后的序列按列写入，然后按行读出。

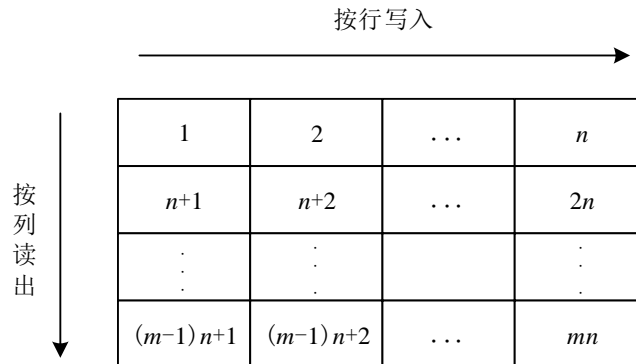


图 2 分组交织器的交织过程

以  $(3, 4)$  分组交织器为例，设输入序列：

$$\mathbf{C}=[c_1 \ c_2 \ c_3 \ c_4 \ c_5 \ c_6 \ c_7 \ c_8 \ c_9 \ c_{10} \ c_{11} \ c_{12}]$$

交织后得到：

$$\tilde{\mathbf{C}}=[c_1 \ c_5 \ c_9 \ c_2 \ c_6 \ c_{10} \ c_3 \ c_7 \ c_{11} \ c_4 \ c_8 \ c_{12}]$$

**卷积交织：**

$(B, M)$  卷积交织器的结构如图 3 所示。输入端与输出端的旋转开关“同步地循环”运行，信息序列按顺序分别进入  $B$  条支路，经过各条支路延迟不同周期后输出：第 1 条支路无延迟，第 2 条支路延迟  $M$  个码元周期，第 3 条支路延迟  $2M$  个码元周期，……，第  $B$  条支路延迟  $(B-1)M$  个码元周期。该卷积交织器的交织深度为  $BM+1$ ，总时延为  $B(B-1)M$ ，一般采用  $B(B-1)M/2$  个移位寄存器实现数据的延迟。

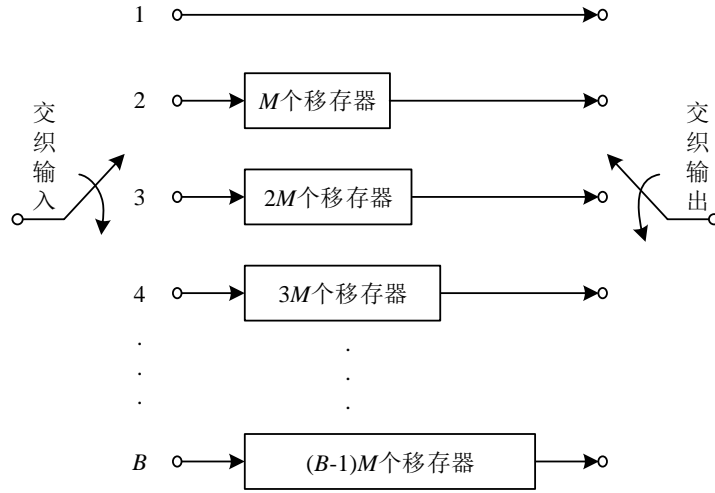


图 3 卷积交织器

去交织器的结构如图 4 所示，其各支路延迟的周期数与交织器相反：第 1 条支路延迟  $(B-1)M$  个码元周期，第 2 条支路延迟  $(B-2)M$  个码元周期，第 3 条支路延迟  $(B-3)M$  个码元周期，……，第  $B$  条支路无延迟。

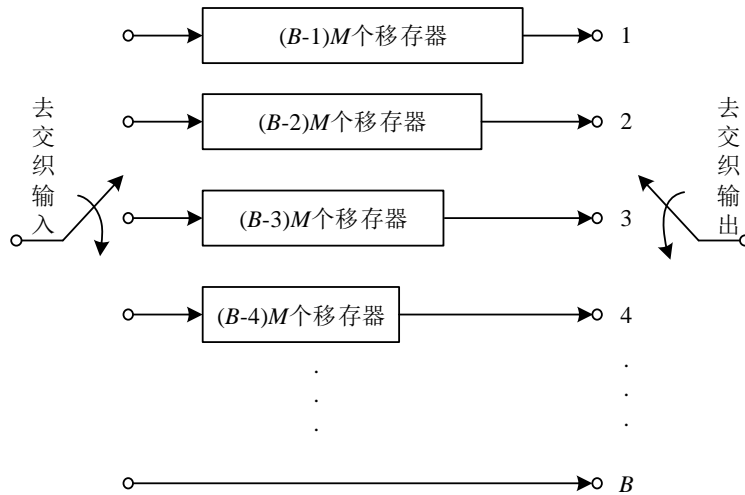


图 4 卷积去交织器

以  $(3, 1)$  卷积交织为例，其交织器与去交织器结构如图 5 所示。输入端与输出端的旋转开关依次同步地连接 3 条延迟支路，信息序列按顺序分别进入支路

中：第 1 条支路无延迟直接输出，第 2 条支路延迟 1 个码元周期后输出，第 3 条支路延迟 2 个码元周期后输出。

交织前序列与交织后序列之间的关系如图 6 所示。

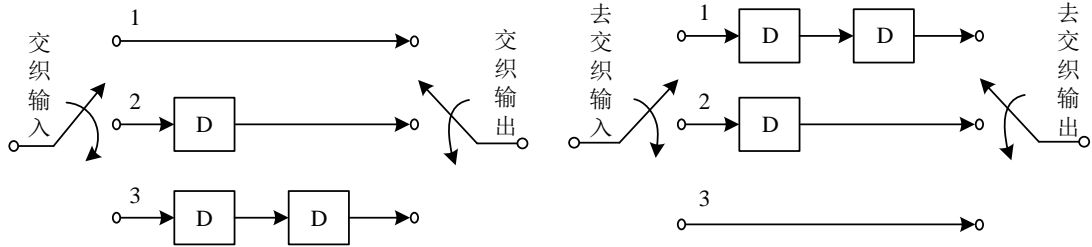


图 5 (3, 1) 卷积交织器与去交织器

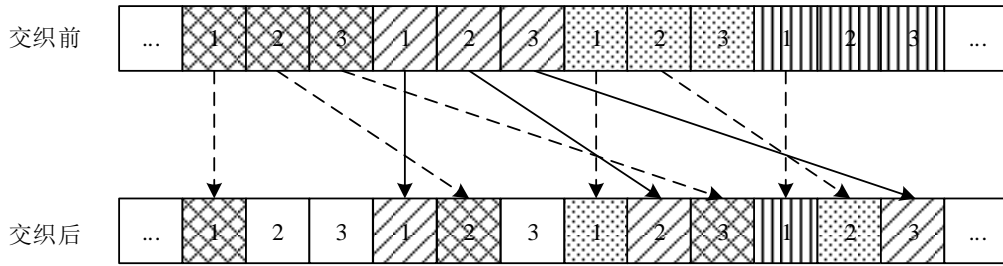


图 6 信息序列经 (3,1) 卷积交织前后的对应关系

设交织前序列：

$$\mathbf{C}=[c_1 \ c_2 \ c_3 \ c_4 \ c_5 \ c_6]$$

交织开始之前将各支路的移寄存器清零。从  $t=0$  到  $t=2$  时刻， $c_1$ 、 $c_2$  和  $c_3$  依次进入支路 1、支路 2 与支路 3：进入支路 1 的  $c_1$  直接输出，而  $c_2$  和  $c_3$  分别进入支路 2 和支路 3 的第一个移寄存器，这两条支路的输出均为 0；从  $t=3$  到  $t=5$  时刻， $c_4$ 、 $c_5$  和  $c_6$  依次进入支路 1、支路 2 与支路 3：进入支路 1 的  $c_4$  直接输出，而  $c_5$  和  $c_6$  分别进入支路 2 和支路 3 的第一个移寄存器，原先在支路 2 第一个移寄存器中的  $c_2$  输出，原先在支路 3 第一个移寄存器中的  $c_3$  移至第二个移寄存器，支路 3 的输出为 0。

为了使移寄存器中的码元全部输出，需对序列进行补零操作。从  $t=6$  到  $t=8$  时刻，各支路的输入均为 0：进入支路 1 的 0 直接输出，原先在支路 2 第一个移寄存器中的  $c_5$  输出，原先在支路 3 第一个移寄存器中的  $c_6$  移至第二个移寄存器，原先在第二个移寄存器中的  $c_3$  输出；从  $t=9$  到  $t=11$  时刻，各支路的输入均为 0：进入支路 1 的 0 直接输出，原先在支路 2 第一个移寄存器中的 0 输出，原先在支路 3 第一个移寄存器中的 0 移至第二个移寄存器，原先在第二个移寄存器中的  $c_6$  输出。

交织后得到：

$$\tilde{\mathbf{C}}=[c_1 \ 0 \ 0 \ c_4 \ c_2 \ 0 \ 0 \ c_5 \ c_3 \ 0 \ 0 \ c_6]$$

**随机交织：**

随机交织器是指交织映射关系随机生成的交织器，每个长度为  $N$  的随机交

织器共有  $N!$  种可能的交织形式。显然，随机交织器未必能带来最优的交织增益；交织长度  $N$  越大，交织器性能越好，设计复杂度越高。

不同结构交织器的对比见表 1。

表 1 不同结构交织器的比较

	$(m,n)$ 分组交织	$(B,M)$ 卷积交织	随机交织 (交织深度 $N$ )
交织深度	$m$	$BM+1$	未知
时延	$mn$	$B(B-1)M$	$N$
占用移存器数目	$2mn$	$B(B-1)M/2$	$2N$

### 三、 实验预习

设交织前的信息序列  $\mathbf{C} = [c_1 \ c_2 \ c_3 \ c_4 \ \dots \ c_{11} \ c_{12}]$ ：

1. 将  $\mathbf{C}$  输入  $(4, 3)$  分组交织器得到交织结果  $\tilde{\mathbf{C}}$ ，说明该交织器的交织深度、时延与占用移存器数目；
2. 将  $\mathbf{C}$  输入  $(4, 1)$  卷积交织器得到交织结果  $\tilde{\mathbf{C}}$ ；说明该交织器的交织深度、时延与占用移存器数目。（交织开始之前交织器的各移存器处于全零状态，交织结束后信息码元应全部移出交织器。）

### 四、 实验内容（要求给出结果截图，源代码放在.m文件中）

1. 给定分组交织矩阵行数  $m=7$ ，列数  $n=4$ ：

（1）编写 MATLAB 函数（function）实现分组交织：函数输入为交织矩阵的行数  $m$ 、列数  $n$  与信息序列  $\mathbf{C}$ ，输出为交织结果  $\tilde{\mathbf{C}}$ 。（考虑到可能出现输入的信息序列无法完整分组的情况，可在序列末尾适当补零。）

当输入信息序列  $\mathbf{C}=[1 \ 2 \ 3 \ \dots \ 19 \ 20]$ 时，输出交织结果。

（2）编写 MATLAB 函数（function）实现分组去交织：函数输入为交织矩阵的行数  $m$ 、列数  $n$  与交织序列  $\tilde{\mathbf{C}}$ ，输出为原序列  $\mathbf{C}$ 。

2. 给定卷积交织的延迟支路数  $B=4$ 、各延迟支路的单位移存器数目  $M=7$ ：

（1）编写 MATLAB 函数（function）实现卷积交织：函数输入为延迟支路数  $B$ 、各延迟支路的单位移存器数目  $M$  与信息序列  $\mathbf{C}$ ，输出为交织结果  $\tilde{\mathbf{C}}$ 。（交织开始之前交织器的各移存器处于全零状态，交织结束后信息码元应全部移出交织器。）

当输入信息序列  $\mathbf{C}=[1 \ 2 \ 3 \ \dots \ 19 \ 20]$ 时，输出交织结果。

(2) 编写 MATLAB 函数 (function) 实现卷积去交织: 函数输入为延迟支路数  $B$ 、各延迟支路的单位移存器数目  $M$  与交织序列  $\tilde{C}$ , 输出为原序列  $C$ 。

3. 给定随机交织长度  $N=28$ :

(1) 编写 MATLAB 函数 (function) 实现随机交织: 函数输入为交织长度  $N$ 、随机映射关系  $s$  与信息序列  $C$ , 输出为交织结果  $\tilde{C}$ 。(参考函数: randperm()。)

当输入信息序列  $C=[1\ 2\ 3\ \dots\ 19\ 20]$ 时, 输出交织结果。

(2) 编写 MATLAB 函数 (function) 实现随机去交织: 函数输入为交织长度  $N$ 、随机映射关系  $s$  与交织序列  $\tilde{C}$ , 输出为原序列  $C$ 。

4. 级联编译码器结构如图 7 所示, 交织器分别采用 (7, 4) 分组交织、(4, 7) 卷积交织和长度  $N=28$  的随机交织。在 AWGN 信道传输与 BPSK 调制的条件下, 绘制这三种不同交织结构的级联码与未编码系统、(7, 4) 汉明编码系统、(2, 1, 3) 卷积编码系统的误码率曲线。信噪比范围: 0~4dB。

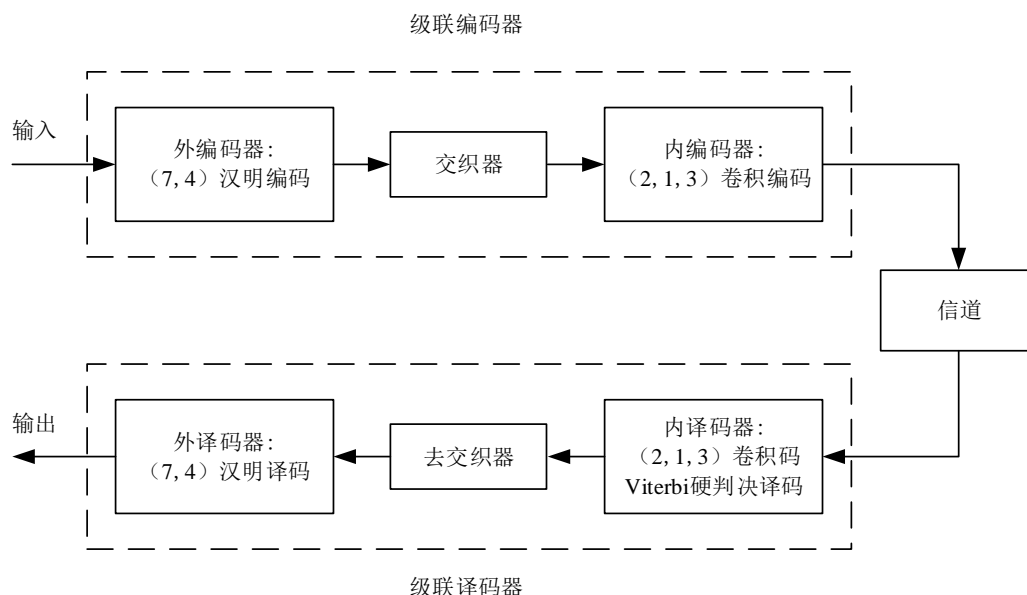


图 7 级联编译码器结构框图

## 五、 实验思考题

1. 对比分析在 AWGN 信道传输与 BPSK 调制的条件下, 不同交织结构的级联码 (外码为 (7, 4) 汉明码, 内码为 (2, 1, 3) 卷积码, 交织结构分别为 (7, 4) 分组交织、(4, 7) 卷积交织与  $N=28$  随机交织) 与未编码系统、(7, 4) 汉明编码系统、(2, 1, 3) 卷积编码系统的误码性能。