

信道编码-MATLAB仿真实验中的应用_v2

殷蔚华

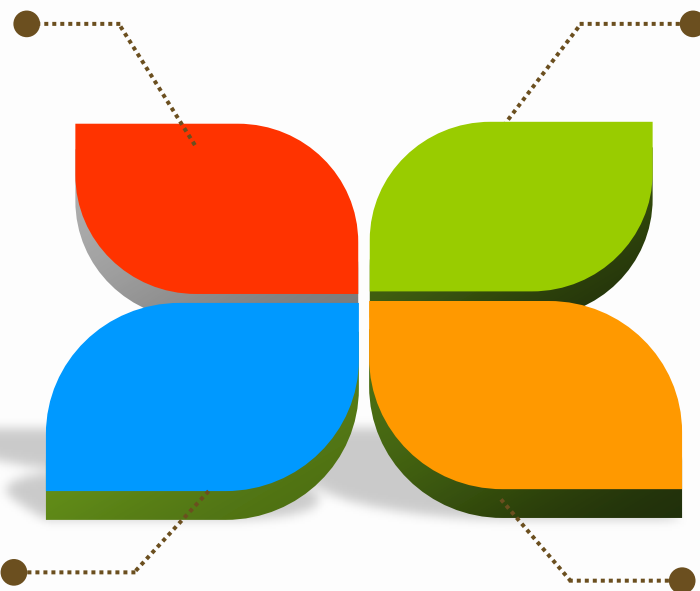
目录

一、信道编码概述

二、卷积码的结构描述

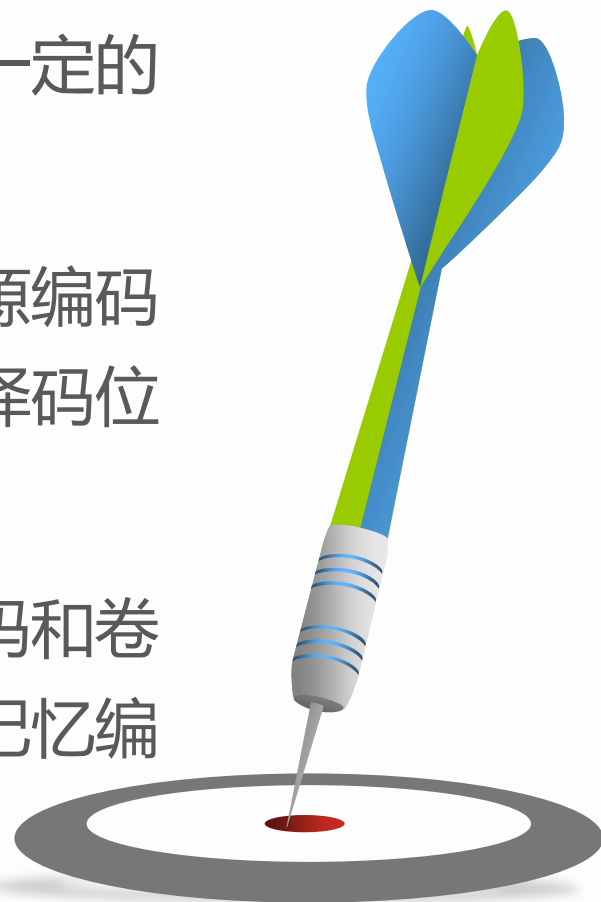
四、卷积码译码

三、卷积码编码



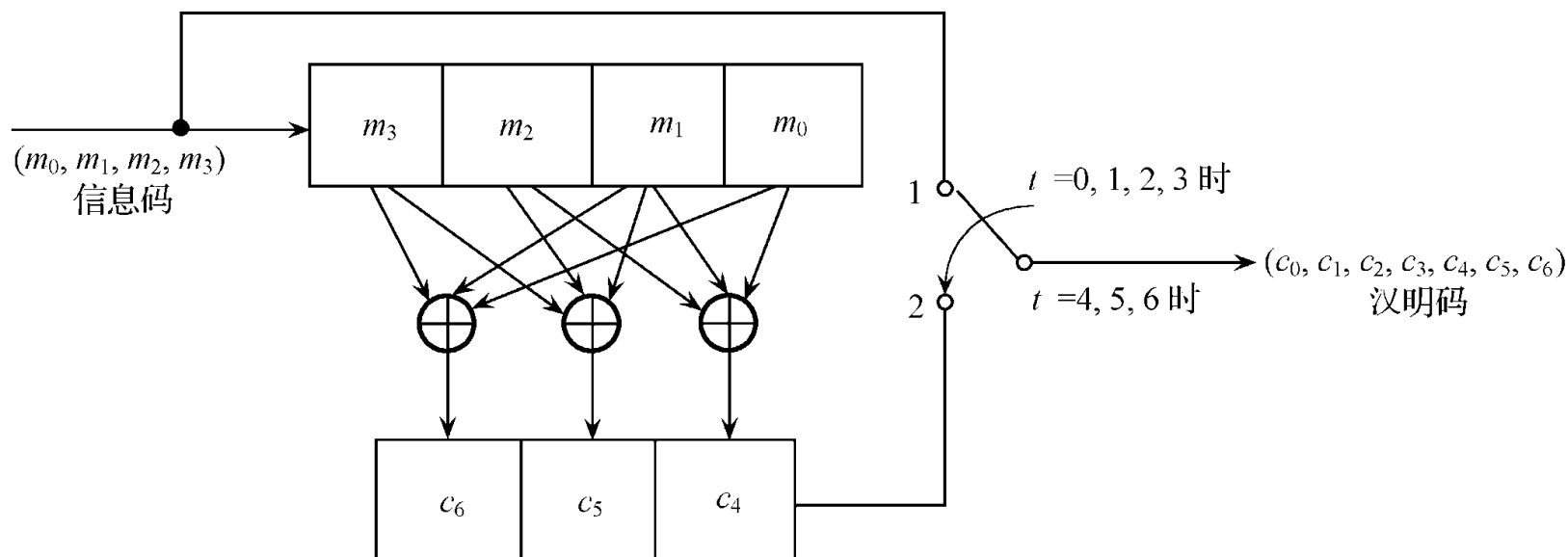
一、信道编码概述

- 信道编码又称检纠错编码，通过增加一定的冗余度以提高数字通信系统的可靠性。
- 在数字通信系统中，信道编码位于信源编码之后，数字调制之前；相应的，信道译码位于数字解调之后，信源译码之前。
- 信道编码的两大基本类型为线性分组码和卷积码，前者为无记忆编码，后者为有记忆编码。



● Example 1: (7, 4) 线性分组码的编码器

- $m_0 \sim m_3$ 为信息码，直接对应 $c_0 \sim c_3$ (系统码)。信息码长 $k=4$;
- $c_4 \sim c_6$ 为加入的冗余位，由 $m_0 \sim m_3$ 的线性组合构成。线性分组码长 $n=7$, $c_0 \sim c_6$ 称为一个**码字**;
- **码率** $= k/n = 4/7$, 称为 **(n, k) 线性分组码**。

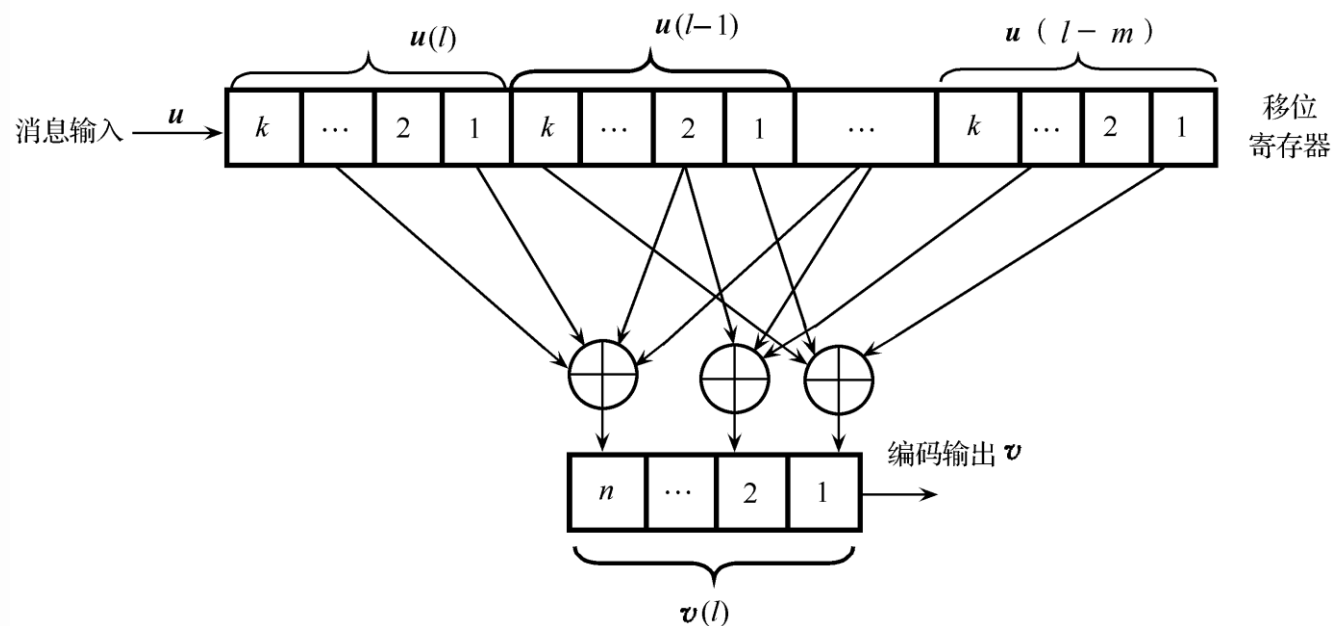


线性分组码的常用概念：

- 两个码字逐位比较，若有 d 个对应位上的值不同，则称两者间码距为 d ；
- 将一个线性分组码的所有码字两两比较，找到的最小码距称为该码的**最小码距 d_{\min}** ；
- d_{\min} 可用于判断该码的检纠错能力，如下：
 - 要求能检测出 l 位错，则需 $d_{\min} \geq l + 1$
 - 要求能纠正 t 位错，则需 $d_{\min} \geq 2t + 1$
 - 要求能检测出 l 位错，并能纠正其中的 t 位错，则需 $d_{\min} \geq l + t + 1$

● Example 2: (n, k, m) 前馈型卷积码的编码器

- 每 k 位信息码输入，相应输出 n 位卷积码；
- 当前 n 位编码输出=当前 k 位信息码的线性组合+此前 m 组的 k 位信息码的线性组合，即有记忆码；
- 输入无限长时，**码率 $\approx k/n$** ，称为 **(n, k, m) 卷积码**。



卷积码的码距与检纠错能力：

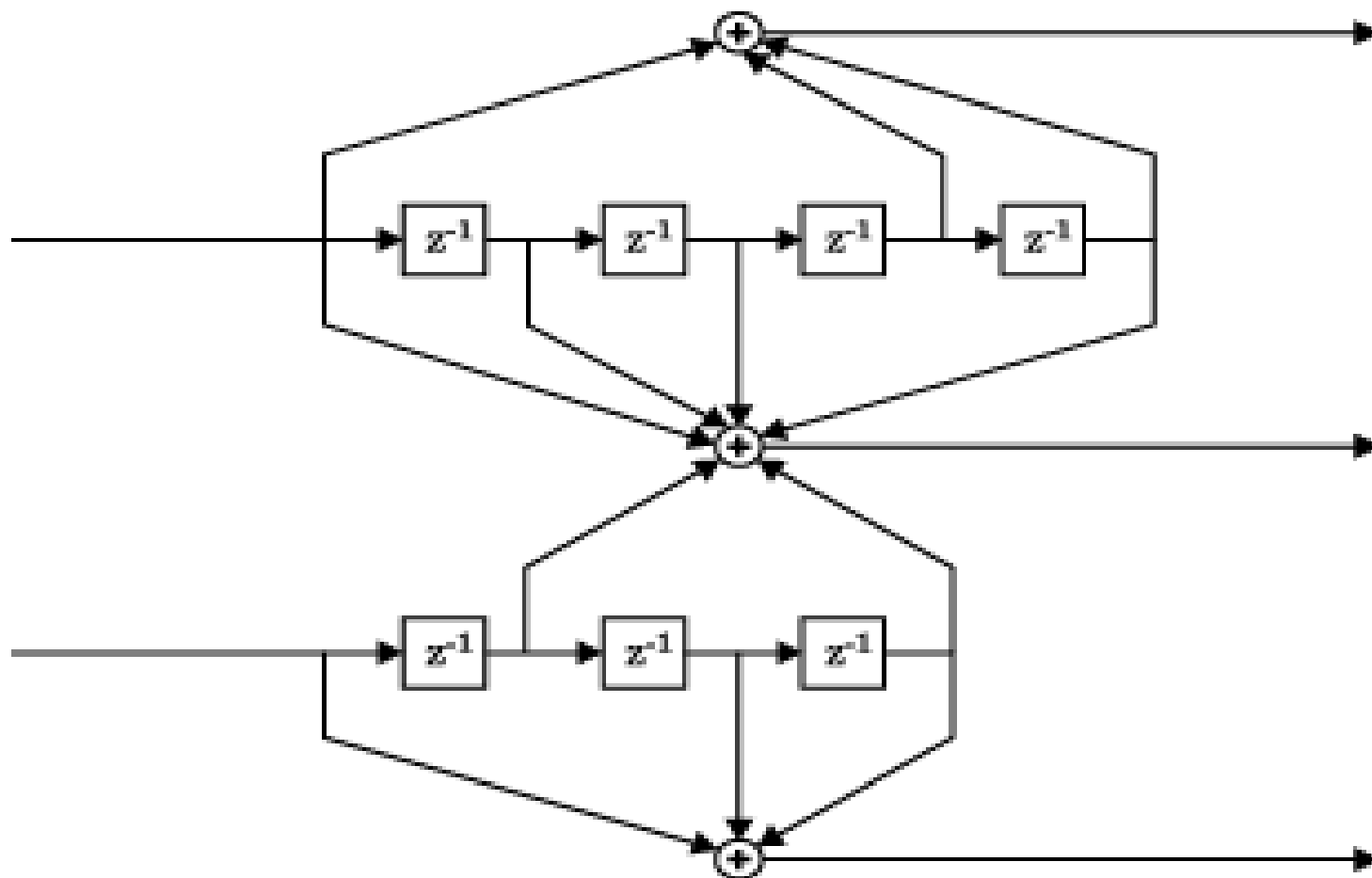
- 卷积码的检纠错能力与码距的关系可套用线性分组码的三个公式；
- 但由于卷积码是有记忆码，其最小码距有几种不同的定义，采用哪一个与选择的译码方式有关；
- 目前广泛使用维特比译码，此时的最小码距称为自由距离 d_f ，定义为码长为无限长时两两码字间的最小距离。计算自由距离 d_f 要用到卷积码的状态图描述，在此略。

-
- MATLAB提供了通信系统仿真的各种常用函数，其中关于信道编码的常用函数在/Communications Toolbox/End-to-End Simulation/ [Error Detection and Correction](#)下。
 - MATLAB中，关于卷积码最常用的是以下几个函数：
 - [poly2trellis](#) 将卷积码的码多项式转换为网格结构
 - [convenc](#) 卷积码的编码（二进制数据）
 - [vitdec](#) 卷积码的维特比译码（二进制数据）

二、卷积码的结构描述

- 卷积码有多种描述方式，在MATLAB中主要使用码多项式和网格结构两种描述方式，主要函数为：
 - poly2trellis 将卷积码的码多项式转换为网格结构
- 最常用的函数格式为：**trellis = poly2trellis(ConstraintLength, CodeGenerator)**
 - 输入参数1——ConstraintLength，约束长度，=编码器中产生延迟的移位寄存器的个数+1；
 - 输入参数2——CodeGenerator，用于表示编码器输入输出关系的码多项式；
 - 输出参数——trellis，用于表示编码器输入输出关系的网格结构。

● **Example 3:** 码率为2/3的前馈型卷积码的编码器结构



- 输入为每符号2bit (u_1, u_2) , 输出为每符号3bit (v_1, v_2, v_3) , 则码率=输入比特数/输出比特数=2/3。

- 上支路的约束长度=4+1=5, 下支路的约束长度=3+1=4,

- 则ConstraintLength=[5 4]。

- 上支路的输入输出关系为: $u_1 \rightarrow v_1: (10011)_2 = (23)_8;$

$$u_1 \rightarrow v_2: (11101)_2 = (35)_8; \quad u_1 \rightarrow v_3: (00000)_2 = (00)_8;$$

- 下支路的输入输出关系为: $u_2 \rightarrow v_1: (0000)_2 = (00)_8;$

$$u_2 \rightarrow v_2: (1101)_2 = (15)_8; \quad u_2 \rightarrow v_3: (1011)_2 = (13)_8;$$

- 则CodeGenerator=[23 35 0; 0 5 13]。

- 最终得到网格结构为:

trellis = poly2trellis([5 4],[23 35 0; 0 5 13])

三、卷积码编码

- 卷积码编码的MATLAB函数为：
 - convenc 卷积码的编码（二进制数据）
- 最常用的函数格式为：**code = convenc(msg,trellis)**
 - 输入参数1——msg，未编码的信息符号序列，二进制矢量形式；
 - 输入参数2——trellis，卷积码编码器的网格结构；
 - 输出参数——code，编码后的卷积码符号序列，二进制矢量形式。

四、卷积码译码

- 卷积码译码的MATLAB函数为：

- vitdec 卷积码的维特比译码（二进制数据）

- 最常用的函数格式为：

`decoded = vitdec(code,trellis,tblen,opmode,dectype);`

`decoded = vitdec(code,trellis,tblen,opmode,'soft',nsdec)`

- 输入参数1——code，维特比译码器的输入符号序列，矢量形式。以前述2/3码率的编码器结构为例，每个符号代表编码器输出的3个bit；
- 输入参数2——trellis，卷积码编码器的网格结构；
- 输入参数3——tblen，a positive integer scalar，用于规定回溯深度。If the code rate is 1/2, a typical value for tblen is about five times the constraint length of the code;

- 输入参数4——[opmode](#)，指示译码器的操作模式，与编码器的结尾处理方式有关。通常取值= 'term'，此时假设编码器是从移位寄存器的全零状态开始，也结束于全零状态。为此，编码时，信息符号输入完毕后，需要额外输入若干0比特，以使移位寄存器恢复为全零状态；
- 输出参数——[decoded](#)，维特比译码器的输出符号序列，矢量形式。以前述2/3码率的编码器结构为例，每个符号代表编码器输入的2个bit。

- 输入参数5—— [dectype](#)，指示译码器的判决类型。其取值不同，对应的输入参数1——code的数据类型也不同。其取值如下表：

Values of dectype Input	Meaning
'unquant'	软判决，code的数据类型为实数（未量化），其中1表示逻辑 '0'，-1表示逻辑 '1'。
'hard'	硬判决，code 的数据类型为二进制。
'soft'	软判决，code 的数据类型为0及正整数（表示不同的量化值）。此时需与输入参数nsdec 一起用，函数格式为 decoded = vitdec(code,trellis,tblen,opmode,'soft',nsdec)

A person is surfing on a large, curling blue wave under a bright blue sky with white clouds. The surfer is in a crouched position, riding the face of the wave. The water is a deep blue, and the wave's crest is white with foam. The overall scene is dynamic and energetic.

感谢聆听，请开始尝试！