

第11章 卷积码

正如第1章指出的,卷积码不同于分组码之处在于:编码器有记忆,且在任意给定时间单元内,编码器的输出不仅取决于该时刻的输入,也依赖于一定数量的以前的输入。一个存储级数为 m 的码率 $R = k/n$ 的卷积编码器,可以用输入存储为 m 的 k 输入、 n 输出线性时序电路实现;换言之,输入在进入编码器后还在其中额外驻留了 m 个时间单元。通常, k 和 n 是小的整数, $k < n$,信息序列被分成长度为 k 的分组,码字被分成长度为 n 的分组。在 $k=1$ 的重要的特例下,信息序列不必分段而可以连续处理。不同于分组码的是,卷积码不是通过增加 k 和 n ,而是通过增加存储级数 m 来实现大的最小距离和低的误码率。

本章讲述卷积码编码过程,以及表示卷积码的方法。我们将说明怎样用状态图表示卷积编码器,以及怎样计算卷积码的权重计数函数。最后,我们介绍卷积码的一些距离测度。

Elias^[1]在1955年时最早引入了不同于分组码的卷积码。稍后,Wozencraft和Reiffen^[2]提出了一种对约束长度大的卷积码的有效的译码方法,即序列译码,并且实验性研究很快开始出现。1963年,Massey^[3]提出了一种较低效率但易于实现的译码方案,称为门限译码。这一进展使卷积码在电话、卫星和无线信道的数字传输中得到了大量实际应用。第13章详细讨论这两种次优译码方法。而后在1967年,维特比(Viterbi)^[4]提出了最大似然(ML)译码算法,它易于对约束长度小的卷积码实现软判决译码。维特比算法,以及软判决序列译码,使卷积码在70年代就用于深空和卫星通信系统中。1974年,Bahl、Cocke、Jelinek和Raviv(BCJR)^[5]对信息比特具有不等先验概率的卷积码提出了最大后验概率(MAP)译码算法。近年来,BCJR算法已应用到软判决迭代译码中,其中信息比特的先验概率在每次迭代中都发生变化。第12章中详细讨论这两种最优译码算法,第16章中介绍相应的迭代译码。

1976年,Ungerboeck和Csajka^[6]引入了网格编码调制的概念,它采用码率 $R = k/(k+1)$ 的短约束长度卷积码及软判决维特比译码。(更详细的介绍参考Ungerboeck^[7]。)这些方案是在带宽受限的电话信道上实现高速数字传输的基础,将在第18章详细讨论。Ungerboeck的工作激发人们了对卷积码反馈编码器性质的兴趣,因为系统反馈形式表示高码率编码最有效。另一个系统反馈卷积编码器应用的例子是turbo编码(turbo coding),它由Berrou、Glavieux和Thitimajshima^[8]在1993年引入。(更完整的介绍参考Berrou和Glavieux^[9]。)这种编码方案将两个系统反馈编码器的并行连接和迭代MAP译码结合起来,可以在SNR接近香农极限时实现相当低的约 10^{-5} 的BER。目前多种应用正在考虑采用这种技术,包括数字蜂窝信道上的数据传输、深空和卫星通信,以及高速数字磁记录,第16章中详细讨论turbo码。

很多编码理论和数字通信方面的书都不同程度地讲述了卷积码。最近出了三本专门讲述卷积码的书^{[10][11][12]}。Johannesson和Zigangirov的书^[12]是最全面的,包含了对卷积码各方面的深入分析。

11.1 卷积码的编码

我们用一系列例子来讲述卷积码编码的基本内容。卷积码的编码器分成两类:前馈和反馈。进一步地,在每种类别中,编码器可以是系统的或非系统的。我们从考虑一类非系

统前馈编码器(nonsystematic feedforward encoders)开始。

例 11-1 码率 $R=1/2$ 的非系统前馈卷积编码器

一个存储级数为 $m=3$ 的二进制码率 $R=1/2$ 非系统前馈卷积编码器的方框图示于图 11-1 中。注意到编码器由 $k=1$ 个延时单元为 $m=3$ 的移位寄存器, 以及 $n=2$ 个模 2 和加法器构成。模 2 和加法器可以用异或门实现。(注意到如图 11-1 的任意多输入模 2 和加法器都可以采用一系列二输入模 2 和加法器, 或者说是异或门来实现。)由于模 2 和是线性运算, 所以编码器是线性系统。

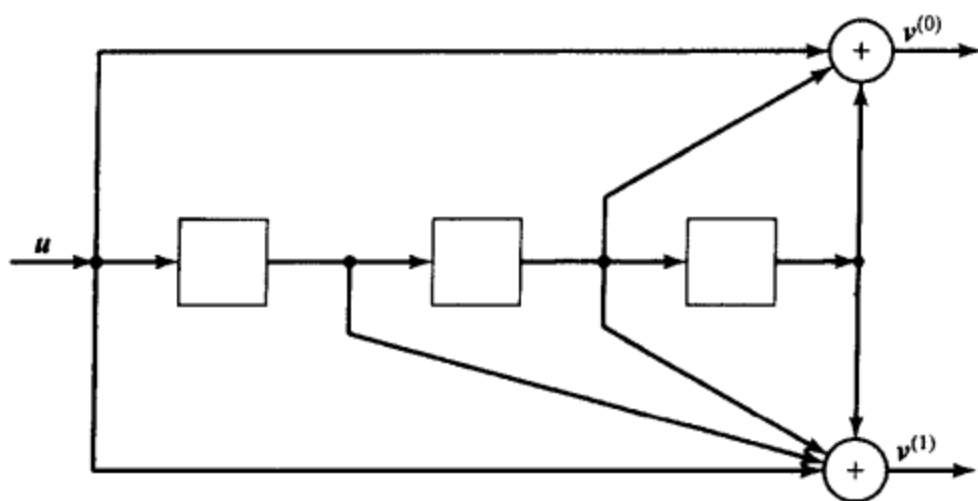


图 11-1 存储级数为 $m=3$ 的码率 $R=1/2$ 二进制非系统前馈卷积编码器

所有卷积码都可以采用这种类型的线性前馈移位寄存器实现。

信息序列 (information sequence) $u = (u_0, u_1, u_2, \dots)$ 每次有一个比特进入编码器。由于编码器是线性系统, 两个编码器输出序列 (output sequence) $v^{(0)} = (v_0^{(0)}, v_1^{(0)}, v_2^{(0)}, \dots)$ 和 $v^{(1)} = (v_0^{(1)}, v_1^{(1)}, v_2^{(1)}, \dots)$ 可以通过输入序列 u 和两个编码器冲激响应 (impulse responses) 的卷积而得到。冲激响应可以通过令 $u = (1\ 0\ 0\ \dots)$ 并观察两个输出序列而得到。对存储级数为 m 的编码器, 脉冲响应最多可以持续 $m+1$ 个时间单元, 因而可以写成 $g^{(0)} = (g_0^{(0)}, g_1^{(0)}, \dots, g_m^{(0)})$ 和 $g^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$ 。对图 11-1 中的编码器

$$g^{(0)} = (1\ 0\ 1\ 1) \quad (11-1a)$$

$$g^{(1)} = (1\ 1\ 1\ 1) \quad (11-1b)$$

冲激响应 $g^{(0)}$ 和 $g^{(1)}$ 称为编码器的生成序列 (generator sequences)。

现在我们将编码方程 (encoding equation) 写成

$$v^{(0)} = u \otimes g^{(0)} \quad (11-2a)$$

$$v^{(1)} = u \otimes g^{(1)} \quad (11-2b)$$

其中, $\otimes$ 表示离散卷积, 所有的运算都是模 2 和。卷积运算意味着对所有的 $l \geq 0$

$$v_l^{(j)} = \sum_{i=0}^m u_{l-i} g_i^{(j)} = u_l g_0^{(j)} + u_{l-1} g_1^{(j)} + \dots + u_{l-m} g_m^{(j)}, \quad j = 0, 1 \quad (11-3)$$

其中对所有 $l < i$ 有 $u_{l-i} \triangleq 0$, 所有的运算都是模 2 和。因此, 对图 11-1 中的编码器

$$v_l^{(0)} = u_l + u_{l-2} + u_{l-3} \quad (11-4a)$$

$$v_l^{(1)} = u_l + u_{l-1} + u_{l-2} + u_{l-3} \quad (11-4b)$$

这很容易通过直接观察编码器图表而进行验证。编码后, 为经信道传输, 两个输出序列合并成单个序列, 称为码字 (codeword)。此码字由下式给出

$$v = (v_0^{(0)} v_0^{(1)}, v_1^{(0)} v_1^{(1)}, v_2^{(0)} v_2^{(1)}, \dots) \quad (11-5)$$

例如, 令信息序列为 $u = (1\ 0\ 1\ 1\ 1)$, 则输出序列是

$$v^{(0)} = (1\ 0\ 1\ 1\ 1) \otimes (1\ 0\ 1\ 1) = (1\ 0\ 0\ 0\ 0\ 0\ 0\ 1) \quad (11-6a)$$

$$v^{(1)} = (1\ 0\ 1\ 1\ 1) \otimes (1\ 1\ 1\ 1) = (1\ 1\ 0\ 1\ 1\ 1\ 0\ 1) \quad (11-6b)$$

码字是

$$v = (1\ 1, 0\ 1, 0\ 0, 0\ 1, 0\ 1, 0\ 1, 0\ 0, 1\ 1) \quad (11-7)$$

如果我们将生成序列 $g^{(0)}$ 和 $g^{(1)}$ 交织, 而后重排成矩阵形式

$$G = \begin{bmatrix} g_0^{(0)} g_0^{(1)} & g_1^{(0)} g_1^{(1)} & g_2^{(0)} g_2^{(1)} & \cdots & g_m^{(0)} g_m^{(1)} \\ & g_0^{(0)} g_0^{(1)} & g_1^{(0)} g_1^{(1)} & \cdots & g_{m-1}^{(0)} g_{m-1}^{(1)} & g_m^{(0)} g_m^{(1)} \\ & & g_0^{(0)} g_0^{(1)} & \cdots & g_{m-2}^{(0)} g_{m-2}^{(1)} & g_{m-1}^{(0)} g_{m-1}^{(1)} & g_m^{(0)} g_m^{(1)} \\ & & & \ddots & & & \ddots \end{bmatrix} \quad (11-8)$$

其中空白地方是全零，则我们可以把编码方程重写成矩阵形式

$$v = uG \quad (11-9)$$

同样地，其中所有的运算是模2和。矩阵 G 称为编码器的(时域)生成矩阵(generator matrix)。注意到， G 的每一行等于前一行，但向右移了 $n=2$ 位，与信息序列 u 可以是任意长度的事实相对应， G 是半无限矩阵。如果 u 有有限长度 h ，则 G 有 h 行和 $2(m+h)$ 列， v 具有长度 $2(m+h)$ 。对图 11-1 中的编码器，如果 $u = (10111)$ ，则

$$v = uG = (1 \ 0 \ 1 \ 1 \ 1) = \begin{bmatrix} 11 & 01 & 11 & 11 \\ & 11 & 01 & 11 & 11 \\ & & 11 & 01 & 11 & 11 \\ & & & 11 & 01 & 11 & 11 \\ & & & & 11 & 01 & 11 & 11 \end{bmatrix}$$

$$= (11, 01, 00, 01, 01, 01, 00, 11) \quad (11-10)$$

和我们前面使用离散卷积的计算结果相同。 ■

例 11.2 码率 $R=2/3$ 非系统前馈卷积编码器

作为非系统前馈卷积编码器的第二个例子，考虑图 11-2 中示的存储级数为 $m=1$ 的二进制码率 $R=2/3$ 编码器。编码器由 $k=2$ 个延时单元为 $m=1$ 的移位寄存器及 $n=3$ 个模2和加法器。信息序列每次有 $k=2$ 个比特进入编码器，可以写成 $u = (u_0, u_1, \dots) = (u_0^{(1)} u_0^{(2)}, u_1^{(1)} u_1^{(2)}, u_2^{(1)} u_2^{(2)}, \dots)$ ，或者两个输入序列 $u^{(1)} = (u_0^{(1)}, u_1^{(1)}, u_2^{(1)}, \dots)$ 和 $u^{(2)} = (u_0^{(2)}, u_1^{(2)}, u_2^{(2)}, \dots)$ 。对应于每个输入序列有三个生成序列。令 $g_i^{(j)} = (g_{i,0}^{(j)}, g_{i,1}^{(j)}, \dots, g_{i,m}^{(j)})$ 代表对应于第 i 个输入和第 j 个输出的生成序列，我们得到图 11-2 中编码器的生成序列

$$g_1^{(0)} = (1 \ 1) \quad g_1^{(1)} = (0 \ 1) \quad g_1^{(2)} = (1 \ 1) \quad (11-11a)$$

$$g_2^{(0)} = (0 \ 1) \quad g_2^{(1)} = (1 \ 0) \quad g_2^{(2)} = (1 \ 0) \quad (11-11b)$$

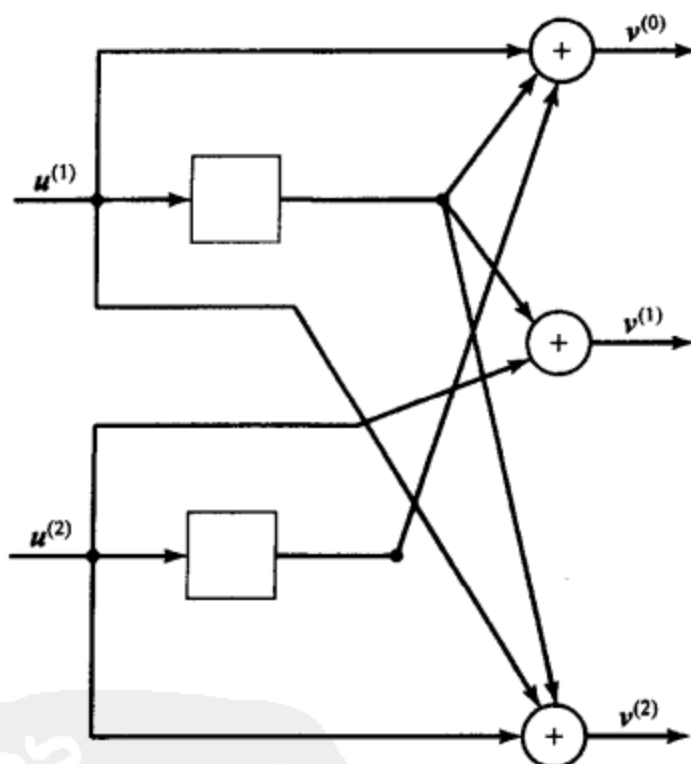


图 11-2 存储级数为 $m=1$ 的码率 $R=2/3$ 二进制非系统前馈卷积编码器

我们将编码方程写成

$$v^{(0)} = u^{(1)} \otimes g_1^{(0)} + u^{(2)} \otimes g_2^{(0)} \quad (11-12a)$$

$$v^{(1)} = u^{(1)} \otimes g_1^{(1)} + u^{(2)} \otimes g_2^{(1)} \quad (11-12b)$$

$$v^{(2)} = u^{(1)} \otimes g_1^{(2)} + u^{(2)} \otimes g_2^{(2)} \quad (11-12c)$$

卷积运算意味着

$$v_i^{(0)} = v_i^{(1)} + u_{i-1}^{(1)} + v_{i-1}^{(2)} \quad (11-13a)$$

$$v_i^{(1)} = v_i^{(2)} + u_{i-1}^{(1)} \quad (11-13b)$$

$$v_i^{(2)} = v_i^{(1)} + u_i^{(2)} + u_{i-1}^{(1)} \quad (11-13c)$$

可以直接从编码器图表中看出。复合之后，码字由下式给出

$$\mathbf{v} = (v_0^{(0)} v_0^{(1)} v_0^{(2)}, v_1^{(0)} v_1^{(1)} v_1^{(2)}, v_2^{(0)} v_2^{(1)} v_2^{(2)}, \dots) \quad (11-14)$$

例如，如果 $\mathbf{u}^{(1)} = (1\ 0\ 1)$ 和 $\mathbf{u}^{(2)} = (1\ 1\ 0)$ ，则

$$\mathbf{v}^{(0)} = (1\ 0\ 1) \otimes (1\ 1) + (1\ 1\ 0) \otimes (0\ 1) = (1\ 0\ 0\ 1) \quad (11-15a)$$

$$\mathbf{v}^{(1)} = (1\ 0\ 1) \otimes (0\ 1) + (1\ 1\ 0) \otimes (1\ 0) = (1\ 0\ 0\ 1) \quad (11-15b)$$

$$\mathbf{v}^{(2)} = (1\ 0\ 1) \otimes (1\ 1) + (1\ 1\ 0) \otimes (1\ 0) = (0\ 0\ 1\ 1) \quad (11-15c)$$

以及

$$\mathbf{v} = (110, 000, 001, 111) \quad (11-16)$$

码率 $R = 2/3$ 编码器的生成矩阵是

$$\mathbf{G} = \begin{bmatrix} g_{1,0}^{(0)} g_{1,0}^{(1)} g_{1,0}^{(2)} & g_{1,1}^{(0)} g_{1,1}^{(1)} g_{1,1}^{(2)} & \cdots & g_{1,m}^{(0)} g_{1,m}^{(1)} g_{1,m}^{(2)} \\ g_{2,0}^{(0)} g_{2,0}^{(1)} g_{2,0}^{(2)} & g_{2,1}^{(0)} g_{2,1}^{(1)} g_{2,1}^{(2)} & \cdots & g_{2,m}^{(0)} g_{2,m}^{(1)} g_{2,m}^{(2)} \\ & g_{1,0}^{(0)} g_{1,0}^{(1)} g_{1,0}^{(2)} & \cdots & g_{1,m-1}^{(0)} g_{1,m-1}^{(1)} g_{1,m-1}^{(2)} & g_{1,m}^{(0)} g_{1,m}^{(1)} g_{1,m}^{(2)} \\ & g_{2,0}^{(0)} g_{2,0}^{(1)} g_{2,0}^{(2)} & \cdots & g_{2,m-1}^{(0)} g_{2,m-1}^{(1)} g_{2,m-1}^{(2)} & g_{2,m}^{(0)} g_{2,m}^{(1)} g_{2,m}^{(2)} \\ & \ddots & & \ddots \end{bmatrix} \quad (11-17)$$

矩阵形式的编码方程同样由 $\mathbf{v} = \mathbf{uG}$ 给出。注意到， $\mathbf{G}$ 的每个 $k=2$ 行组等于前一个行组，但向右移了 $n=3$ 位。对图 11-2 中的编码器，如果 $\mathbf{u}^{(1)} = (101)$ 和 $\mathbf{u}^{(2)} = (110)$ ，则 $\mathbf{u} = (\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2) = (11, 01, 10)$ ，并且

$$\mathbf{v} = \mathbf{uG} = (11, 01, 10) \begin{bmatrix} 101 & 111 \\ 011 & 100 \\ & 101 & 111 \\ & 011 & 100 \\ & & 101 & 111 \\ & & & 011 & 100 \end{bmatrix} = (110, 000, 001, 111) \quad (11-18)$$

这又和前面我们采用离散卷积的计算结果一致。 ■

在图 11-1 和图 11-2 中，从存储输入序列的 k 个移位寄存器到形成输出序列的 n 个模 2 和加法器之间的连线，分别直接对应于公式 (11-1) 和 (11-11) 给出的 kn 个生成序列中的非零项。同样地，任意 kn 个生成序列的集合都可以用来实现和图 11-1 和 11-2 所示相同类型的码率 $R = k/n$ 前馈卷积编码器，它采用 k 个移位寄存器存储输入序列， n 个模 2 和加法器形成输出序列。特别地， k 个输入序列从每个移位寄存器的左端进入， n 个输出序列由移位寄存器外面的模 2 和加法器产生。这种实现称为控制器规范形式 (controller canonical form)，任何前馈卷积编码器可以用这种方式实现。(在本节的稍后我们将会看到，反馈编码器也可以用控制器规范形式实现。)

例 11-2 明显表明，编码以及表示它的方法在输入序列数目 $k > 1$ 时更加复杂。尤其是，编码器包含 k 个移位寄存器，但它们并非需要有相同的长度。我们现在给出以控制器规范形式实现的卷积编码器中使用的有关移位寄存器长度的定义。

定义 11-1 令 v_i 为有 k 个输入序列的卷积编码器中第 i 个移位寄存器的长度， $i = 1, 2, \dots, k$ 。

例如, 对图 11-1 中的码率 $R = 1/2$ 编码器, $v = 3$; 对图 11-2 中的码率 $R = 2/3$ 编码器, $v_1 = v_2 = 1$ 。

定义 11-2 编码器存储级数 (memory order) m 定义为

$$m = \max_{1 \leq i \leq k} v_i \quad (11-19)$$

即, m 是所有 k 个移位寄存器长度的最大值。

对图 11-1 中的码率 $R = 1/2$ 编码器, $m = v_1 = 3$; 对图 11-2 中的码率 $R = 2/3$ 编码器, $m = \max(v_1, v_2) = \max(1, 1) = 1$ 。

定义 11-3 编码器的整体约束长度 (overall constraint length) v 定义如下

$$v = \sum_{1 \leq i \leq k} v_i \quad (11-20)$$

即, v 是所有 k 个移位寄存器的长度之和。

对图 11-1 中的码率 $R = 1/2$ 编码器, $v = v_1 = 3$; 对图 11-2 中的码率 $R = 2/3$ 编码器, $v = v_1 + v_2 = 1 + 1 = 2$ 。

定义 11-4 具有整体约束长度 v 的码率 $R = k/n$ 卷积编码器称为 (n, k, v) 编码器。图 11-1 中的码率 $R = 1/2$ 编码器是 $(2, 1, 3)$ 编码器, 图 11-2 中的码率 $R = 2/3$ 编码器是 $(3, 2, 2)$ 编码器。

注意到对一个 $(n, 1, v)$ 编码器, 整体约束长度 v 等于存储级数 m , 即当 $k = 1$ 时 $v = m$; 但是对于 $k > 1$ 的情形, 通常 $m \leq v$, 当每个移位寄存器有相同的长度时, $v = km$ 。

例 11-3 $(4, 3, 3)$ 非系统前馈卷积编码器

一个 $(4, 3, 3)$ 二进制非系统前馈卷积编码器示于图 11-3 中, 其移位寄存器长度是 $v_1 = 0$, $v_2 = 1$ 和 $v_3 = 2$ 。存储级数是 $m = 2$, 整体约束长度是 $v = 3$ 。仿照图 11-2 中的方法, 生成序列由下式给出

$$\mathbf{g}_1^{(0)} = (100) \quad \mathbf{g}_1^{(1)} = (100) \quad \mathbf{g}_1^{(2)} = (100) \quad \mathbf{g}_1^{(3)} = (100) \quad (11-21a)$$

$$\mathbf{g}_2^{(0)} = (000) \quad \mathbf{g}_2^{(1)} = (110) \quad \mathbf{g}_2^{(2)} = (010) \quad \mathbf{g}_2^{(3)} = (100) \quad (11-21b)$$

$$\mathbf{g}_3^{(0)} = (000) \quad \mathbf{g}_3^{(1)} = (010) \quad \mathbf{g}_3^{(2)} = (101) \quad \mathbf{g}_3^{(3)} = (101) \quad (11-21c)$$

对一般情形的存储级数为 m 的 (n, k, v) 前馈编码器, 生成矩阵由下式给出

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \mathbf{G}_2 & \cdots & \mathbf{G}_m \\ & \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_{m-1} & \mathbf{G}_m \\ & & \mathbf{G}_0 & \cdots & \mathbf{G}_{m-2} & \mathbf{G}_{m-1} & \mathbf{G}_m \\ & & & \ddots & & & \end{bmatrix} \quad (11-22)$$

其中每个 $\mathbf{G}_l$ 是 $k \times n$ 子矩阵, 其项为

$$\mathbf{G}_l = \begin{bmatrix} g_{1,l}^{(0)} & g_{1,l}^{(1)} & \cdots & g_{1,l}^{(n-1)} \\ g_{2,l}^{(0)} & g_{2,l}^{(1)} & \cdots & g_{2,l}^{(n-1)} \\ \vdots & \vdots & & \vdots \\ g_{k,l}^{(0)} & g_{k,l}^{(1)} & \cdots & g_{k,l}^{(n-1)} \end{bmatrix} \quad (11-23)$$

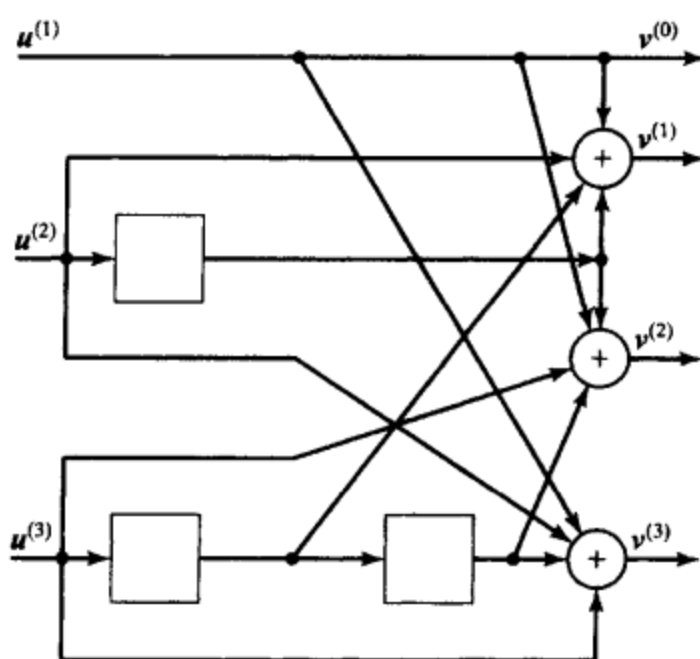


图 11-3 $(4, 3, 3)$ 二进制非系统前馈卷积编码器

再一次注意到, G 的每个 k 行组等于前一个行组, 但向右移了 n 位。对信息序列 $u = (u_0, u_1, \dots) = (u_0^{(1)} u_0^{(2)} \dots u_0^{(k)}, u_1^{(1)} u_1^{(2)} \dots u_1^{(k)}, \dots)$, 其码字 $v = (v_0, v_1, \dots) = (v_0^{(0)} v_0^{(1)} \dots v_0^{(n-1)}, v_1^{(0)} v_1^{(1)} \dots v_1^{(n-1)}, \dots)$ 由 $v = uG$ 给出。现在我们应该给出卷积码的正式定义了。

定义 11-5 一个 (n, k, v) 卷积码是由一个 (n, k, v) 卷积编码器产生的所有输出序列 (码字) 的集合, 即它是编码器生成矩阵 G 的行空间。

由于码字 v 是生成矩阵 G 的行的线性组合, (n, k, v) 卷积码是线性码。 ■

例 11-3 (续)

由图 11-3 中 $(4, 3, 3)$ 卷积编码器产生的所有码字的集合是一个 $(4, 3, 3)$ 卷积码。 ■

在任何线性系统中, 涉及卷积的时域运算可以被更方便的涉及多项式乘法的变换域运算代替。由于卷积码是线性系统, 编码方程中的每个序列可以用相应的多项式代替, 并且卷积运算可以用多项式相乘代替。在二进制序列的多项式表示中, 序列本身由多项式的系数表示。例如, 对一个 $(2, 1, v)$ 编码器, 编码方程成为

$$v^{(0)}(D) = u(D)g^{(0)}(D) \quad (11-24a)$$

$$v^{(1)}(D) = u(D)g^{(1)}(D) \quad (11-24b)$$

其中

$$u(D) = u_0 + u_1 D + u_2 D^2 + \dots \quad (11-25a)$$

是信息序列,

$$v^{(0)}(D) = v_0^{(0)} + v_1^{(0)} D + v_2^{(0)} D^2 + \dots \quad (11-25b)$$

和

$$v^{(1)}(D) = v_0^{(1)} + v_1^{(1)} D + v_2^{(1)} D^2 + \dots \quad (11-25c)$$

是编码序列,

$$g^{(0)}(D) = g_0^{(0)} + g_1^{(0)} D + \dots + g_m^{(0)} D^m \quad (11-25d)$$

和

$$g^{(1)}(D) = g_0^{(1)} + g_1^{(1)} D + \dots + g_m^{(1)} D^m \quad (11-25e)$$

是码的生成多项式(generator polynomials), 所有的运算都是模 2 和。我们可以将码字写成

$$V(D) = [v^{(0)}(D), v^{(1)}(D)] \quad (11-26a)$$

或者在复合之后为

$$v(D) = v^{(0)}(D^2) + Dv^{(1)}(D^2) \quad (11-26b)$$

变量 D 可以解释成延时操作, 其中 D 的幂表示一个比特相对序列中初始比特的延时时间单元数。 ■

例 11-1 (续)

对图 11-1 中的 $(2, 1, 3)$ 编码器, 生成多项式是 $g^{(0)}(D) = 1 + D^2 + D^3$ 和 $g^{(1)}(D) = 1 + D + D^2 + D^3$ 。对信息序列 $u(D) = 1 + D^2 + D^3 + D^4$, 编码方程是

$$v^{(0)}(D) = (1 + D^2 + D^3 + D^4)(1 + D^2 + D^3) = 1 + D^7 \quad (11-27a)$$

$$v^{(1)}(D) = (1 + D^2 + D^3 + D^4)(1 + D + D^2 + D^3) = 1 + D + D^3 + D^4 + D^5 + D^7 \quad (11-27b)$$

我们可以将码字写成

$$V(D) = [1 + D^7, 1 + D + D^3 + D^4 + D^5 + D^7] \quad (11-28a)$$

或

$$v(D) = 1 + D + D^3 + D^7 + D^9 + D^{11} + D^{14} + D^{15} \quad (11-28b)$$

注意到每种情形下, 该结果和使用卷积及矩阵相乘得到的结果相同。 ■

用控制器规范形式实现的编码器的生成矩阵可以直接根据编码器方框图确定, 正如前面

对生成序列指出的。从移位寄存器的延时元素到输出(模2加法器)的连线序列(1代表连接, 0代表不连接)是相应生成多项式中的系数序列, 即它是生成序列。生成多项式的最低次(常数)项对应和移位寄存器左端的连线, 最高次项对应于和移位寄存器右端的连接。例如, 图11-1中从移位寄存器到输出 $v^{(0)}$ 的连线的序列是 $g^{(0)} = (1011)$, 对应生成多项式为 $g^{(0)}(D) = 1 + D^2 + D^3$ 。

由于用控制器规范形式实现的 $(n, 1, v)$ 编码器右端的移位寄存器必须至少连接到一个输出, 至少有一个生成多项式的次数等于移位寄存器长度 $v_1 = v = m$, 即

$$m = \max_{0 \leq j \leq n-1} [\deg g^{(j)}(D)] \quad (11-29)$$

在一个 $k > 1$ 的 (n, k, v) 编码器中, 对 k 个输入的每一个都有 n 个生成多项式。 n 个生成器的每个集合代表从一个移位寄存器到 n 个输出的连线序列, 因此

$$v_i = \max_{0 \leq j \leq n-1} [\deg g_i^{(j)}(D)], \quad 1 \leq i \leq k \quad (11-30)$$

其中 $g_i^{(j)}(D)$ 是对应第 i 个输入到第 j 个输出的生成多项式。

由于编码器是一个线性系统, $u^{(i)}(D)$ 代表第 i 个输入序列, $v^{(j)}(D)$ 代表第 j 个输出序列, 生成多项式 $g_i^{(j)}(D)$ 可以解释成对应从第 i 个输入到第 j 个输出的编码器传递函数。对于任意 k 输入、 n 输出的线性系统, 总共有 kn 个传递函数。它们可以用一个 $k \times n$ (变换域) 生成矩阵表示

$$G(D) = \begin{bmatrix} g_1^{(0)}(D) & g_1^{(1)}(D) & \cdots & g_1^{(n-1)}(D) \\ g_2^{(0)}(D) & g_2^{(1)}(D) & \cdots & g_2^{(n-1)}(D) \\ \vdots & \vdots & & \vdots \\ g_k^{(0)}(D) & g_k^{(1)}(D) & \cdots & g_k^{(n-1)}(D) \end{bmatrix} \quad (11-31)$$

使用生成矩阵, 我们可以把一个 (n, k, v) 前馈编码器的(变换域)编码方程表示成

$$V(D) = U(D)G(D) \quad (11-32)$$

其中 $U(D) \triangleq [u^{(1)}(D), u^{(2)}(D), \dots, u^{(k)}(D)]$ 是输入序列的 k 维向量, $V(D) \triangleq [v^{(0)}(D), v^{(1)}(D), \dots, v^{(n-1)}(D)]$ 是输出序列的 n 维向量, 或者说是码字。复合后, 码字也可以表示成

$$v(D) = v^{(0)}(D^n) + Dv^{(1)}(D^n) + \cdots + D^{n-1}v^{(n-1)}(D^n) \quad (11-33) \blacksquare$$

例 11-2 (续)

对图 11-2 中的 $(3, 2, 2)$ 编码器,

$$G(D) = \begin{bmatrix} 1+D & D & 1+D \\ D & 1 & 1 \end{bmatrix} \quad (11-34)$$

对输入序列 $u^{(1)}(D) = 1 + D^2$ 和 $u^{(2)}(D) = 1 + D$, 编码方程给出码字

$$\begin{aligned} V(D) &= [v^{(0)}(D), v^{(1)}(D), v^{(2)}(D)] = [1 + D^2, 1 + D] \begin{bmatrix} 1+D & D & 1+D \\ D & 1 & 1 \end{bmatrix} \\ &= [1 + D^3, 1 + D^3, D^2 + D^3] \end{aligned} \quad (11-35a)$$

也可以写成

$$v(D) = 1 + D + D^8 + D^9 + D^{10} + D^{11} \quad (11-35b)$$

同样地, 这些结果和采用卷积和矩阵相乘计算得到的相同。 $\blacksquare$

我们可以重写公式(11-31)、(11-32)和(11-33), 以便直接用输入序列来表示复合的码字 $v(D)$ 。稍稍进行代数运算得到

$$v(D) = \sum_{i=1}^k u^{(i)}(D^n) g_i(D) \quad (11-36)$$

其中

$$g_i(D) \triangleq g_i^{(0)}(D^n) + Dg_i^{(1)}(D^n) + \cdots + D^{n-1}g_i^{(n-1)}(D^n), \quad 1 \leq i \leq k \quad (11-37)$$

是对应于第 i 个输入序列到 $v(D)$ 的复合生成多项式 (composite generator polynomial)。

例 11-1 (续)

对图 11-1 中的 (2, 1, 3) 编码器, 复合生成多项式是

$$g(D) = g^{(0)}(D^2) + Dg^{(1)}(D^2) = 1 + D + D^3 + D^4 + D^5 + D^6 + D^7 \quad (11-38)$$

对于 $u(D) = 1 + D^2 + D^3 + D^4$, 码字是

$$\begin{aligned} v(D) &= u(D^2)g(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5 + D^6 + D^7) \\ &= 1 + D + D^3 + D^7 + D^9 + D^{11} + D^{14} + D^{15} \end{aligned} \quad (11-39)$$

再一次地和前面的计算相符。

卷积编码器的一个重要子类是系统编码器 (systematic encoders) 类。在系统编码器中, k 个输出序列, 称为系统输出序列 (systematic output sequences), 正好等于 k 个输入序列。尽管一般来说, k 个系统输出序列可以是任意 k 个输出序列的集合, 但按照最初 (first) k 个输出序列是系统序列的习惯, 即

$$v^{(i-1)} = u^{(i)}, \quad i = 1, 2, \dots, k \quad (11-40)$$

生成序列满足

$$g_i^{(j)} = \begin{cases} 1 & \text{if } j = i - 1 \\ 0 & \text{if } j \neq i - 1 \end{cases} \quad i = 1, 2, \dots, k \quad (11-41)$$

对于系统前馈编码器 (systematic feedforward encoders) 的情形, (时域) 生成矩阵由下式给出

$$G = \begin{bmatrix} IP_0 & 0P_1 & 0P_2 & \cdots & 0P_m \\ & IP_0 & 0P_1 & \cdots & 0P_{m-1} & 0P_m \\ & & IP_0 & \cdots & 0P_{m-2} & 0P_{m-1} & 0P_m \\ & & & \ddots & & & \ddots \end{bmatrix} \quad (11-42)$$

其中 I 是 $k \times k$ 单位矩阵, 0 是 $k \times k$ 全零矩阵, P_l 是 $k \times (n-k)$ 矩阵, 其项为

$$P_l = \begin{bmatrix} g_{1,l}^{(k)} & g_{1,l}^{(k+1)} & \cdots & g_{1,l}^{(n-1)} \\ g_{2,l}^{(k)} & g_{2,l}^{(k+1)} & \cdots & g_{2,l}^{(n-1)} \\ \vdots & \vdots & & \vdots \\ g_{k,l}^{(k)} & g_{k,l}^{(k+1)} & \cdots & g_{k,l}^{(n-1)} \end{bmatrix} \quad (11-43)$$

类似地, $k \times n$ (变换域) 生成序列变成

$$G(D) = \begin{bmatrix} 1 & 0 & \cdots & 0 & g_1^{(k)}(D) & \cdots & g_1^{(n-1)}(D) \\ 0 & 1 & \cdots & 0 & g_2^{(k)}(D) & \cdots & g_2^{(n-1)}(D) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & g_k^{(k)}(D) & \cdots & g_k^{(n-1)}(D) \end{bmatrix} \quad (11-44)$$

由于最初的 k 个输出序列是系统的, 即它们等于 k 个输入序列, 因此它们也可以称为输出信息序列 (output information sequences), 最后的 $n-k$ 个输出序列称为输出校验序列 (output parity sequences)。注意到, 为定义一个 (n, k, v) 非系统前馈编码器通常必须指定 kn 个生成多项式, 然而对系统前馈编码器只需指定 $k(n-k)$ 个多项式。因此, 系统前馈编码器代表所有可能编码器集合的一个子类。任何编码器, 只要其 (变换域) 生成矩阵 $G(D)$ 不在某 k 列组中包含单位矩阵, 则称为非系统的 (nonsystematic)。

对类似公式 (11-42) 或公式 (11-44) 生成矩阵的系统前馈编码器表示的码, 我们可以直接确定其系统奇偶校验矩阵。(时域) 校验矩阵给出如下

$$H = \begin{bmatrix} P_0^T & I & & & & & \\ P_1^T & 0 & P_0^T & I & & & \\ P_2^T & 0 & P_1^T & 0 & P_0^T & I & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ P_m^T & 0 & P_{m-1}^T & 0 & P_{m-2}^T & 0 & \cdots P_0^T & I \\ & & P_m^T & 0 & P_{m-1}^T & 0 & \cdots P_1^T & 0 & P_0^T & I \\ & & & P_m^T & 0 & \cdots P_2^T & 0 & P_1^T & 0 & P_0^T & I \\ & & & & & & & & & \ddots \end{bmatrix} \quad (11-45)$$

式中, I 是 $(n-k) \times (n-k)$ 单位矩阵, 0 是 $(n-k) \times (n-k)$ 全零矩阵。类似地, $(n-k) \times n$ (变换域) 奇偶校验矩阵给出如下

$$H(D) = \begin{bmatrix} g_1^{(k)}(D) & g_2^{(k)}(D) & \cdots & g_k^{(k)}(D) & 1 & 0 & \cdots & 0 \\ g_1^{(k+1)}(D) & g_2^{(k+1)}(D) & \cdots & g_k^{(k+1)}(D) & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ g_1^{(n-1)}(D) & g_2^{(n-1)}(D) & \cdots & g_k^{(n-1)}(D) & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (11-46a)$$

式中, $H(D)$ 的最后 $(n-k)$ 列形成 $(n-k) \times (n-k)$ 单位矩阵。公式(11-46a)中的奇偶校验矩阵可以重写成

$$H(D) = \begin{bmatrix} h_1^{(0)}(D) & h_1^{(1)}(D) & \cdots & h_1^{(k-1)}(D) & 1 & 0 & \cdots & 0 \\ h_2^{(0)}(D) & h_2^{(1)}(D) & \cdots & h_2^{(k-1)}(D) & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ h_{n-k}^{(0)}(D) & h_{n-k}^{(1)}(D) & \cdots & h_{n-k}^{(k-1)}(D) & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (11-46b)$$

式中, 我们定义对应于第 $(i-1)$ 个输出信息序列的第 $(j-k+1)$ 个奇偶校验多项式为 $h_{j-k+1}^{(i-1)}(D) = g_i^{(j)}(D)$, 其中 $j = k, k+1, \dots, n-1$ 和 $i = 1, 2, \dots, k$ 。(在码率 $R = (n-1)/n$ 的重要特例下, 即 $(n-k) = 1$, 我们记不带下标的 $k = n-1$ 个奇偶校验多项式为 $h^{(j)}(D)$, $j = 0, 1, \dots, k-1$ 。)最后, 我们把奇偶校验序列 $h_i^{(j)}$ 和奇偶校验多项式 $h_i^{(j)}(D)$ 的系数相对应起来。

任意码字 v (或 $v(D)$) 必须满足奇偶校验方程

$$vH^T = 0 \quad (11-47)$$

式中矩阵 0 有半无限维数, 或者

$$V(D)H^T(D) = 0(D) \quad (11-48)$$

式中 $0(D)$ 表示全零序列的 $1 \times (n-k)$ 矩阵。换言之, 所有码字 v (或 $v(D)$) 集合是奇偶校验矩阵 H (或 $H(D)$) 的零空间。■

例 11-4 码率 $R = 1/2$ 系统前馈卷积编码器

考虑图 11-4 所示的 $(2, 1, 3)$ 系统前馈编码器。生成序列是 $g^{(0)} = (1000)$ 和 $g^{(1)} = (1101)$, 生成矩阵由下式给出

$$G = \begin{bmatrix} 11 & 01 & 00 & 01 \\ & 11 & 01 & 00 & 01 \\ & & 11 & 01 & 00 & 01 \\ & & & \ddots & & \ddots \end{bmatrix} \quad (11-49)$$

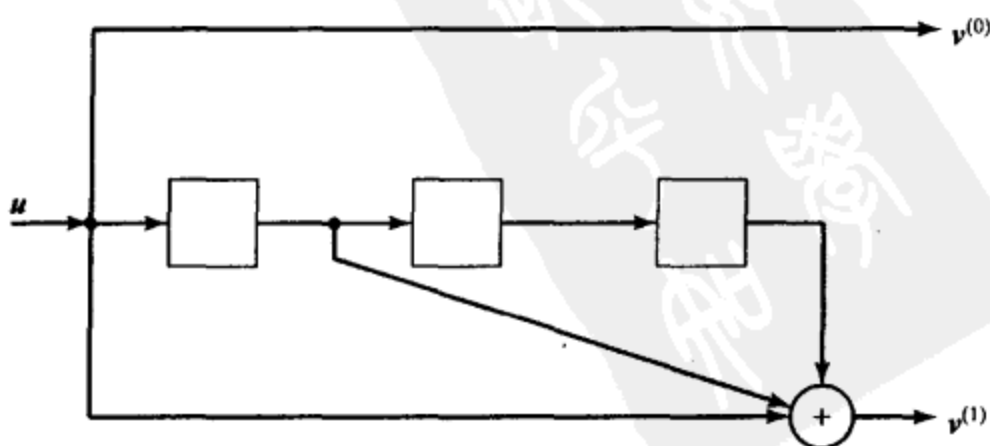


图 11-4 $(2, 1, 3)$ 二进制系统前馈卷积编码器

和

$$\mathbf{G}(D) = [1 \quad 1 + D + D^3] \quad (11-50)$$

对输入序列 $u(D) = 1 + D^2 + D^3$, 输出信息序列是

$$\mathbf{v}^{(0)}(D) = \mathbf{u}(D)\mathbf{g}^{(0)}(D) = (1 + D^2 + D^3)(1) = 1 + D^2 + D^3 \quad (11-51)$$

输出校验序列是

$$\begin{aligned} \mathbf{v}^{(1)}(D) &= \mathbf{u}(D)\mathbf{g}^{(1)}(D) = (1 + D^2 + D^3)(1 + D + D^3) \\ &= 1 + D + D^2 + D^3 + D^4 + D^5 + D^6 \end{aligned} \quad (11-52)$$

码字由下式给出

$$\mathbf{V}(D) = [1 + D^2 + D^3 \quad 1 + D + D^2 + D^3 + D^4 + D^5 + D^6] \quad (11-53a)$$

或者

$$\begin{aligned} \mathbf{v}(D) &= \mathbf{v}^{(0)}(D^2) + D\mathbf{v}^{(1)}(D^2) \\ &= 1 + D + D^3 + D^4 + D^5 + D^6 + D^7 + D^9 + D^{11} + D^{13} \end{aligned} \quad (11-53b)$$

也可以写成

$$\mathbf{v} = (11, 01, 11, 11, 01, 01, 01) \quad (11-53e)$$

使用公式(11-45)和公式(11-46b), 我们可以看到奇偶校验矩阵由下式给出

$$H = \begin{bmatrix} 11 & & & & & \\ 10 & 11 & & & & \\ 00 & 10 & 11 & & & \\ 10 & 00 & 10 & 11 & & \\ & 10 & 00 & 10 & 11 & \\ & & 10 & 00 & 10 & 11 \\ & & & \ddots & & \ddots \end{bmatrix} \quad (11-54)$$

和

$$\mathbf{H}(D) = [\mathbf{h}^{(0)}(D) \quad 1] = [1 + D + D^3 \quad 1] \quad (11-55)$$

对公式(11-53)的码字, 我们有

$$\nu H^T = [11,01,11,11,01,01,01] \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ & 1 & 1 & 0 & 1 \\ & 1 & 0 & 0 & 0 \\ & & 1 & 1 & 0 & 1 \\ & & 1 & 0 & 0 & 0 \\ & & & 1 & 1 & 0 & 1 \\ & & & 1 & 0 & 0 & 0 \\ & & & & 1 & 1 & 1 \\ & & & & & 1 & 0 & 1 \\ & & & & & & 1 & 1 \end{bmatrix} = 0 \quad (11-56)$$

或者

$$\mathbf{V}(D)\mathbf{H}^T(D) = \begin{bmatrix} 1 + D^2 + D^3 & 1 + D + D^2 + D^3 + D^4 + D^5 + D^6 \end{bmatrix} \begin{bmatrix} 1 + D + D^3 \\ 1 \end{bmatrix} = \mathbf{0}(D) \quad (11-57) \blacksquare$$

例 11-5 码率 $R=2/3$ 系统前馈卷积编码器

现在考虑码率 $R=2/3$ 系统前馈编码器，其生成矩阵为

$$G(D) = \begin{bmatrix} 1 & 0 & 1 + D + D^2 \\ 0 & 1 & 1 + D \end{bmatrix} \quad (11-58)$$

使用公式(11-46b)，我们可以将奇偶校验矩阵写成

$$H(D) = [h^{(0)}(D) \quad h^{(1)}(D) \quad 1] = [1 + D + D^2 \quad 1 + D \quad 1] \quad (11-59)$$

编码器的控制器规范形式实现需要总共 $v = v_1 + v_2 = 3$ 个延时单元，如图 11-5a 所示。实现导致了一个(3, 2, 3)编码器。此外，由于输出信息序列由 $v^{(0)}(D) = u^{(1)}(D)$ 和 $v^{(1)}(D) = u^{(2)}(D)$ 给出，则输出奇偶序列由下式给出

$$\begin{aligned} v^{(2)}(D) &= u^{(1)}(D)g_1^{(2)}(D) + u^{(2)}(D)g_2^{(2)}(D) \\ &= (1 + D + D^2)u^{(1)}(D) + (1 + D)u^{(2)}(D) \end{aligned} \quad (11-60)$$

编码器也可以如图 11-5b 所示的实现。注意到这种实现仅需要两个延时单元，而不是三个。

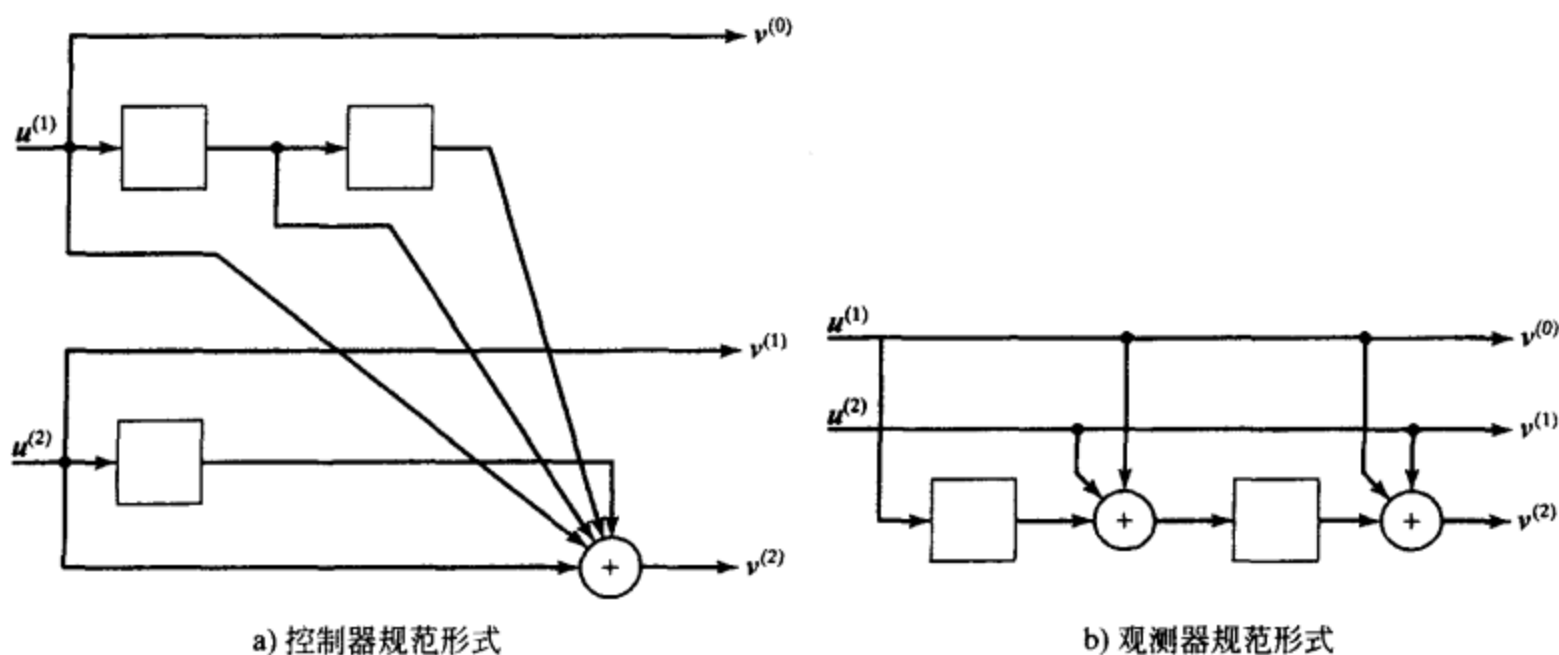


图 11-5 码率 $R=2/3$ 二进制系统前馈卷积编码器的实现

图 11-5b 的编码器实现称为观测器规范形式 (observer canonical form)。通常，对观测器规范形式编码器实现， n 个输出序列中的每一个都对应一个移位寄存器， k 个输入序列进入移位寄存器内部的模 2 和加法器，在每个移位寄存器的右端形成 n 个输出序列。同样，生成(奇偶校验)多项式中的最低次(常数)项代表移位寄存器右端的连线，最高次项代表移位寄存器左端的连线。必须注意到，这正好和控制器规范形式实现中多项式系数和延时元素之间的对应关系相反。基于这个原因，当编码器用观测器规范形式实现时，通常用和习惯相反的次序来写生成(奇偶校验)多项式，即从最高次到最低次。对于如图 11-5b 中的系统编码器，仅有 $(n-k)$ 个移位寄存器， k 个输入序列直接作为前面的 k 个编码器输出序列。在观测器规范形式实现的卷积编码器中使用的有关移位寄存器长度的定义可以类似于定义 11-1 至 11-3 给出。不同之处是在这种情形下，通常存在 n 个移位寄存器而不是 k 个， v_j 是在 $j=0, 1, \dots, n-1$ 范围内定义的。任何前馈卷积编码器都可以用观测器规范形式实现，不久我们将会看到反馈编码器也可以用这种方式实现。因此，控制器规范形式和观测器规范形式提供了两种可以应用于所有卷积编码器的通用实现方法。

例 11-5 (续)

为得到观测器规范形式实现，我们把公式(11-58)的生成矩阵重写成

$$G(D) = \begin{bmatrix} 1 & 0 & D^2 + D + 1 \\ 0 & 1 & D + 1 \end{bmatrix} \quad (11-61)$$

以及公式(11-59)的奇偶校验矩阵写成

$$H(D) = [D^2 + D + 1 \quad D + 1 \quad 1] \quad (11-62)$$

这样, 我们可以清楚地看到公式(11-61)和(11-62)中的多项式系数和图 11-5b 中的连线是一致的。对这种实现, 我们看到 $v_0 = 0$, $v_1 = 0$, $v_2 = 2$, $m = \max_{(0 \leq j \leq n-1)} v_j = 2$ 和 $v = \sum_{(0 \leq j \leq n-1)} v_j = 2$ 。因此, 观测器规范形式实现导致了一个(3, 2, 2)编码器。

例 11-1 引发了一个有趣的问题。两个编码器实现都产生了相同的码, 即 $G(D)$ 的行空间, 但由于两个不同编码器实现的存储要求不同, 那么它是(3, 2, 3)码还是(3, 2, 2)码? 如果我们用控制器规范形式实现编码器, 整体约束长度是 $v = 3$, 导致一个(3, 2, 3)编码器; 但是, 如果我们用观测器规范形式实现编码器, 则 $v = 2$, 导致一个(3, 2, 2)编码器。在下一节我们会看到由于卷积编码器可以用线性时序电路实现, 它可以用包含 2^v 个状态的状态图描述。同样地, 在下一章我们将看到, 卷积码的最大似然译码和最大后验概率译码都要求一个和编码器状态图中状态数成正比的译码复杂度。因此, 我们总是寻找具有最小状态数的编码器实现, 即最小整体约束长度 v 。

我们将在引入另一类系统编码器之后继续对最小编码器实现(minimal encoder realizations)的讨论。系统反馈编码器产生和非系统前馈编码器相同的码, 但是采用了不同的信息序列和码字间的映射关系。我们再一次用一些例子来说明其基本概念。

例 11-6 码率 $R = 1/3$ 系统反馈卷积编码器

考虑码率 $R = 1/3$ 非系统前馈卷积编码器, 其生成矩阵

$$G(D) = [g^{(0)}(D) \quad g^{(1)}(D) \quad g^{(2)}(D)] = [1 + D + D^2 \quad 1 + D^2 \quad 1 + D] \quad (11-63a)$$

其(3, 1, 2)控制器规范形式实现示于图 11-6a 中。颠倒公式(11-63a)中的多项式次序, 我们有

$$G(D) = [D^2 + D + 1 \quad D^2 + 1 \quad D + 1] \quad (11-63b)$$

由此我们得到图 11-6b 所示的(3, 1, 5)观测器规范形式的编码器实现。注意到, 这里控制器规范形式实现有整体约束长度 $v = 2$, 即 4 个状态, 而观测器规范形式实现需要 32 个状态! 现在, 我们用多项式 $g^{(0)}(D) = 1 + D + D^2$ 除 $G(D)$ 中的每一项, 得到修正生成矩阵

$$\begin{aligned} G'(D) &= [1 \quad g^{(1)}(D)/g^{(0)}(D) \quad g^{(2)}(D)/g^{(0)}(D)] \\ &= [1 \quad (1 + D^2)/(1 + D + D^2) \quad (1 + D)/(1 + D + D^2)] \end{aligned} \quad (11-64)$$

用系统反馈形式表示。我们注意到, 这里用 $G'(D)$ 表示的编码器冲激响应有无限长持续时间, 即 $G'(D)$ 的反馈移位寄存器实现是一个无限冲激响应(infinite impulse response, IIR)线性系统。基于这个原因, 对应 $G'(D)$ 的(时域)生成矩阵 G' 包含无限长序列。例如, 式(11-64)中的生成多项式之比 $g^{(1)}(D)/g^{(0)}(D)$ 产生幂级数 $(1 + D^2)/(1 + D + D^2) = 1 + D + D^2 + D^4 + D^5 + D^7 + D^8 + D^{10} + \dots$, 其等价时域形式是无限长序列(11101101101...)。

由 $G'(D)$ 产生的码, 即编码器输出序列(码字)集合, 和由 $G(D)$ 产生的码正好一样。这可以通过下面这点而看出: 在由 $G(D)$ 产生的码中, 如果信息序列 $u(D)$ 产生码字 $V(D)$, 即 $V(D) = u(D)G(D)$, 则在由 $G'(D)$ 产生的码中, 信息序列 $u'(D) = (1 + D + D^2)u(D)$ 产生相同的码字 $V(D)$, 即 $V(D) = u'(D)G'(D)$ 。换言之, $G(D)$ 和 $G'(D)$ 的行空间是相同的, 因此它们产生相同的码。必须注意到, 这两种情形下信息序列和码字间有不同的映射。图 11-6c 显示了对应于公式(11-64)的系统反馈编码器, 用控制器规范形式表示的 4 状态(3, 1, 2)编码器。与此相反, $G'(D)$ 的观测器规范形式实现需要 16 个状态(见习题 11-8)。

最后, 由于 $G'(D)$ 是系统的, 我们可以使用公式(11-46a)来得到奇偶校验矩阵

$$H'(D) = \begin{bmatrix} (1 + D^2)/(1 + D + D^2) & 1 & 0 \\ (1 + D)/(1 + D + D^2) & 0 & 1 \end{bmatrix} \quad (11-65)$$

式中, $H'(D)$ 对 $G(D)$ 和 $G'(D)$ 都是有效的奇偶校验矩阵, 即所有在 $G(D)$ 和 $G'(D)$ 行空间中的码字也在 $H'(D)$ 的零空间中, 并且类似于对生成序列所指出的, 对应 G' 的系统(时域)奇偶校验矩阵 H' 包含无限长度序列。

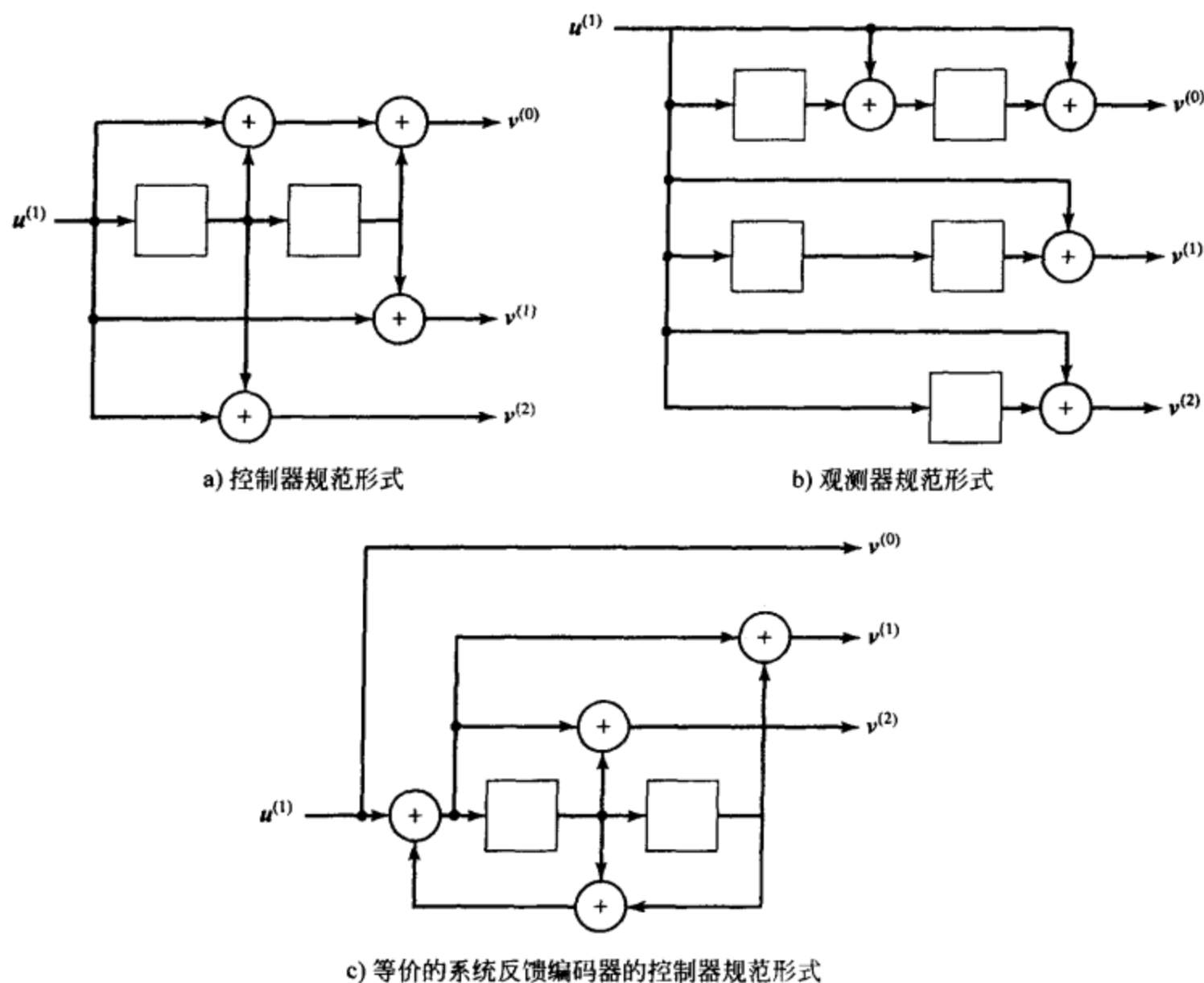


图 11-6 非系统前馈编码器的实现

通常, 如果 $k \times n$ 多项式矩阵 $G(D)$ 是某个码率 $R = k/n$ 非系统卷积编码器的生成矩阵, 则对 $G(D)$ 进行基本(多项式)行运算可以把它转化成一个系统 $k \times n$ 生成矩阵 $G'(D)$, 其项是延时算子 D 的有理函数(rational function), 并且它的最初 k 列形成 $k \times k$ 单位矩阵。由于初等行变换不改变矩阵的行空间, 矩阵 $G(D)$ 和 $G'(D)$ 产生相同的码, 且由于 $G'(D)$ 包含有理函数, 因而它导致了反馈编码器实现。同样地, 由于 $G'(D)$ 是系统形式, 类似于我们在公式(11-46a)中对系统前馈编码器寻找奇偶校验矩阵一样, 它可以用来确定 $(n - k) \times n$ (变换域)系统奇偶校验矩阵 $H'(D)$ 。这里 $H'(D)$ 包含有理函数, 其最后 $(n - k)$ 列形成 $(n - k) \times (n - k)$ 单位矩阵。由于码是 $H'(D)$ 的零空间, 且初等行变换不改变矩阵的零空间, 我们可以把 $H'(D)$ 转化成等价的只包含多项式项的非系统奇偶校验矩阵 $H(D)$ 。

例 11-6 中的生成和奇偶校验序列表明了 IIR 行为是反馈编码器的典型特征。基于这种原因, 反馈编码器更容易使用(变换域)各项包含有理函数的矩阵 $G'(D)$ 和 $H'(D)$ 描述, 而不是相应的各项包含无限长序列的时域矩阵 G' 和 H' 。由于反馈编码器对单个非零输入的响应为无限持续时间, 因此由反馈编码器生成的码称为递归卷积码(recursive convolutional

code)。这样,我们说一个系统反馈编码器产生一个系统递归卷积码(systematic recursive convolutional code, SRCC)。必须注意到,由系统反馈编码器产生的 SRCC 和由等价的非系统前馈编码器产生的非递归卷积码正好包含相同的码字集,但是正如例 11-6 指出的,两种情形下信息序列和码字间的映射是不同的。特别地,对非递归码,重量为 1 的信息序列产生有限长度码字,然而对递归码,重量为 1 的信息序列产生无限长度码字。递归码的这种性质成为第 16 章中要讨论的一类并行级联卷积码,或者 turbo 码实现优秀性能的关键因素。 ■

例 11-2 (续)

公式(11-34)给出了例 11-2 中的非系统前馈生成矩阵 $G(D)$, 图 11-2 显示了其 4 状态 (3, 2, 2) 控制器规范形式编码器实现。(注意到公式(11-34)的观测器规范形式实现需要 8 个状态。)为把 $G(D)$ 转化成等价的系统反馈编码器,我们依次进行下面的初等行变换:

第 1 步 $[1/(1+D)]$ 乘以第 1 行作为新的第 1 行。

第 2 步 第 2 行 $+ [D]$ 乘以第 1 行作为新的第 2 行。

第 3 步 $[(1+D)/(1+D+D^2)]$ 乘以第 2 行作为新的第 2 行。

第 4 步 第 1 行 $+ [D/(1+D)]$ 乘以第 2 行作为新的第 1 行。

得到修正的生成矩阵(见习题 11-7)

$$G'(D) = \begin{bmatrix} 1 & 0 & 1/(1+D+D^2) \\ 0 & 1 & (1+D^2)/(1+D+D^2) \end{bmatrix} \quad (11-66)$$

用系统反馈形式表示。现在,我们可以使用公式(11-46b)来形成系统奇偶校验矩阵

$$\begin{aligned} H'(D) &= [h^{(0)}(D)/h^{(2)}(D) \quad h^{(1)}(D)/h^{(2)}(D) \quad 1] \\ &= [1/(1+D+D^2) \quad (1+D^2)/(1+D+D^2) \quad 1] \end{aligned} \quad (11-67a)$$

或者等价的非系统多项式奇偶校验矩阵

$$H(D) = [h^{(0)} \quad h^{(1)}(D) \quad h^{(2)}(D)] = [1 \quad 1+D^2 \quad 1+D+D^2] \quad (11-67b)$$

式中, $h^{(j)}(D)$ 代表对应第 j 个输出序列的奇偶校验多项式。再一次地,由 $G'(D)$ 生成的递归码和由 $G(D)$ 生成的非递归码正好相同,图 11-7a 显示了其控制器规范形式编码器图,它对应于公式(11-66)的系统反馈编码器。我们注意到,这里图 11-7a 代表 16 状态 (3, 2, 4) 编码器,即在控制器规范形式实现中需要两个长度为 2 的移位寄存器。颠倒公式(11-66)和 (11-67a) 中的多项式次序,我们得到

$$G'(D) = \begin{bmatrix} 1 & 0 & 1/(D^2+D+1) \\ 0 & 1 & (D^2+1)/(D^2+D+1) \end{bmatrix} \quad (11-68)$$

和

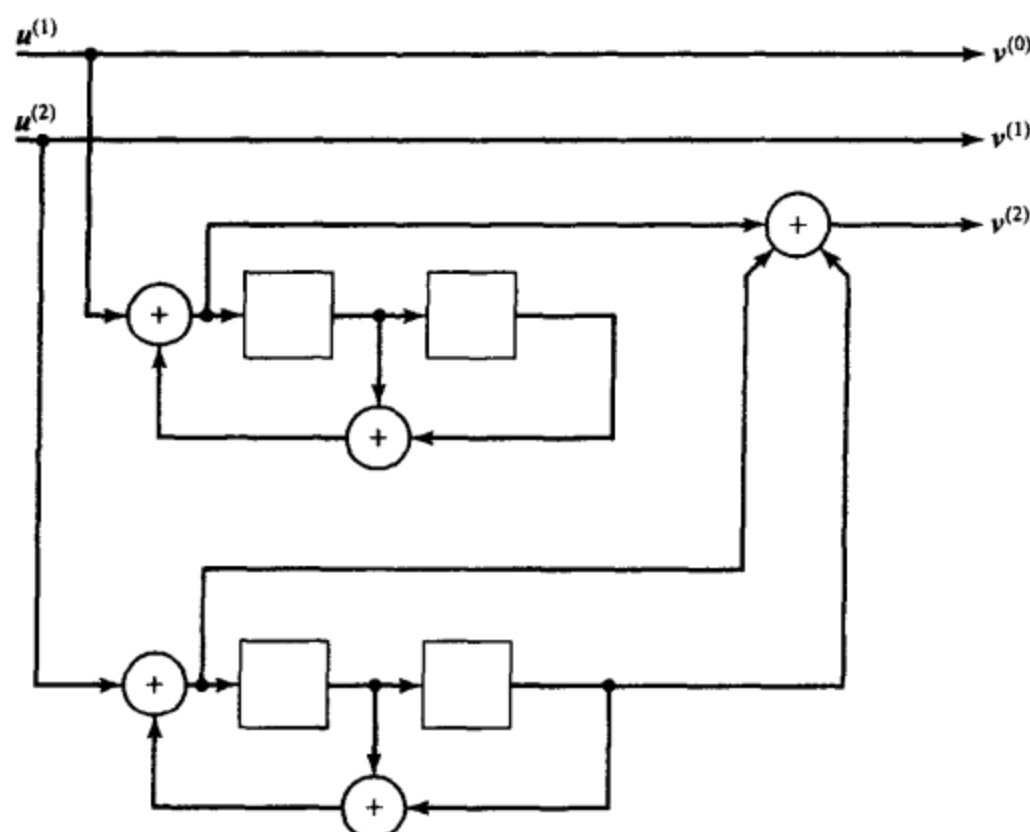
$$H'(D) = [1/(D^2+D+1) \quad (D^2+1)/(D^2+D+1) \quad 1] \quad (11-69)$$

它导致了图 11-7b 所示的观测器规范形式系统反馈编码器实现。我们注意到这代表一个 4 状态 (3, 2, 2) 编码器,即对系统反馈编码器,观测器规范形式实现需要的状态数比控制器规范形式少。 ■

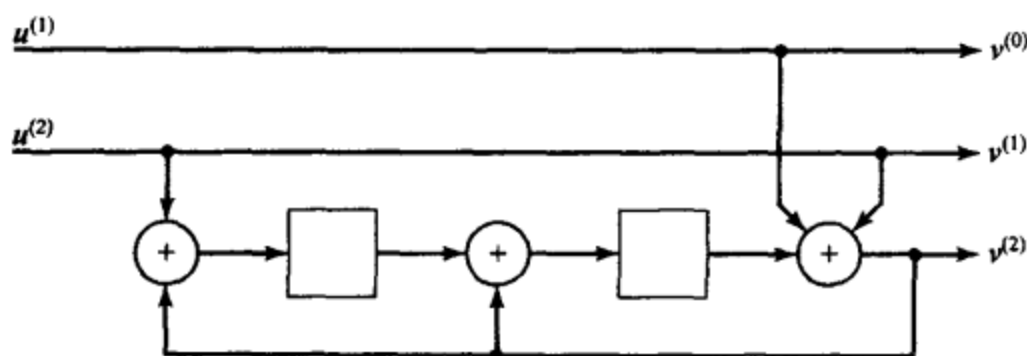
例 11-2 中的码率 $R=2/3$ 码有一个控制器规范形式表示的 4 状态非系统前馈编码器实现(参考图 11-2)。它也有一个观测器规范形式的 4 状态系统反馈编码器实现(见图 11-7b)。因此,这种情形下两个编码器实现都有整体约束长度 $v=2$,即我们得到以控制器规范形式和观测器规范形式表示的两个最小编码器实现。我们现在给出一个码率 $R=2/3$ 码的例子,其中只有观测器规范形式得到最小编码器实现。 ■

例 11-7 码率 $R=2/3$ 系统反馈卷积编码器

现在考虑由下式给出的码率 $R=2/3$ 非系统前馈生成矩阵



a) 控制器规范形式表示的(3, 2, 4)系统反馈编码器



b) 等价的观测器规范形式表示的(3, 2, 2)系统反馈编码器

图 11-7 控制器规范形式编码器图和观测器规范形式系统反馈编码器的实现

$$G(D) = \begin{bmatrix} 1 & D & 1+D \\ 0 & 1+D+D^2+D^3 & 1+D^2+D^3 \end{bmatrix} \quad (11-70)$$

图 11-8a 显示了其 16 状态(3, 2, 4)控制器规范形式编码器实现。(注意到公式(11-70)的观测器规范形式实现需要 64 个状态。)为把 $G(D)$ 转化成等价的系统反馈编码器, 我们依次进行下面的初等行变换:

第 1 步 $[1/(1+D+D^2+D^3)]$ 乘以第 2 行作为新的第 2 行。

第 2 步 第 1 行 $+ [D]$ 乘以第 2 行作为新的第 1 行。

得到修正的生成矩阵(见习题 11-7)

$$G'(D) = \begin{bmatrix} 1 & 0 & (1+D+D^2)/(1+D+D^2+D^3) \\ 0 & 1 & (1+D+D^3)/(1+D+D^2+D^3) \end{bmatrix} \quad (11-71)$$

以系统反馈形式表示, 我们用公式(11-46b)得到系统奇偶校验矩阵

$$\begin{aligned} H'(D) &= [h^{(0)}(D)/h^{(2)}(D) \quad h^{(1)}(D)/h^{(2)}(D) \quad 1] \\ &= [(1+D+D^2)/(1+D+D^2+D^3) \quad (1+D+D^3)/(1+D+D^2+D^3) \quad 1] \end{aligned} \quad (11-72a)$$

或者等价的非系统多项式奇偶校验矩阵

$$H(D) = [1+D+D^2 \quad 1+D+D^3 \quad 1+D+D^2+D^3] \quad (11-72b)$$

这里, 控制器规范形式表示的编码器图对应于公式(11-71)中的系统反馈编码器, 它代

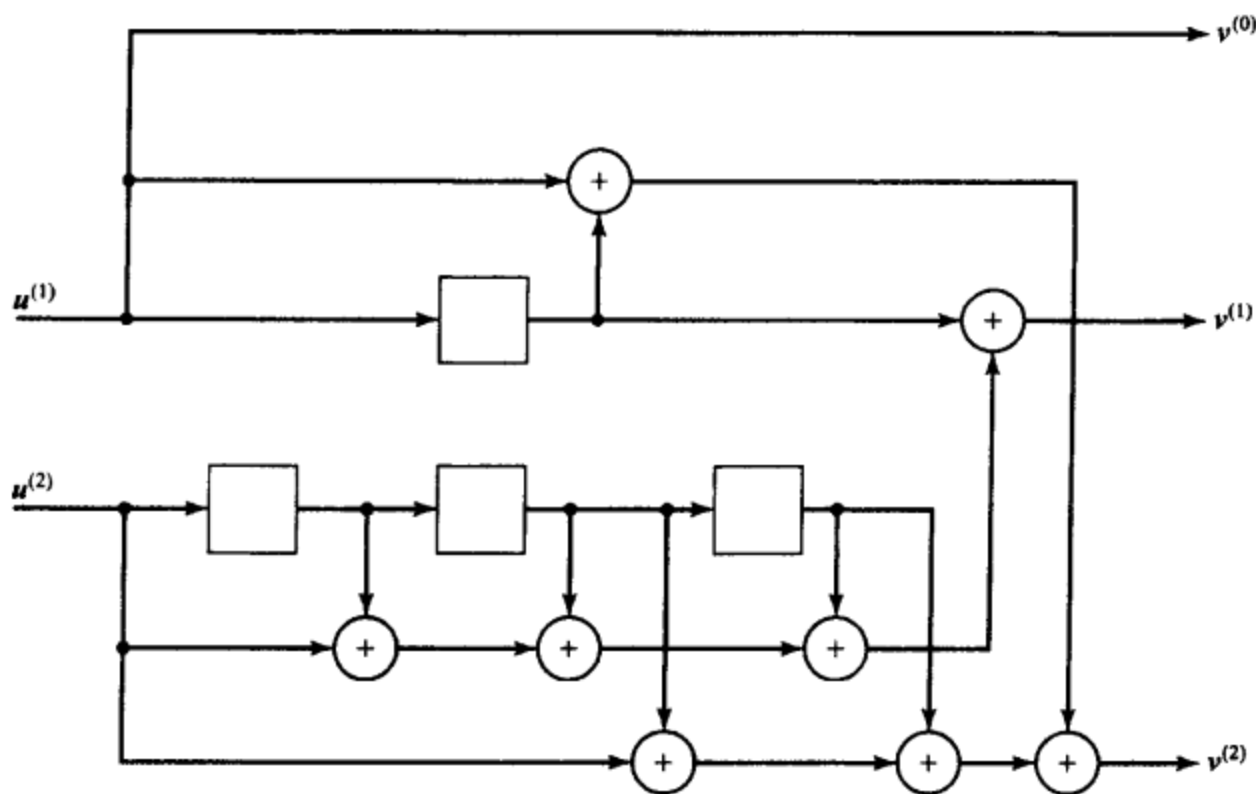
表了一个64状态(3, 2, 6)编码器, 即在控制器规范形式实现中需要两个长度为3的移位寄存器。颠倒公式(11-71)和(11-72a)中的多项式次序, 得到

$$\mathbf{G}'(D) = \begin{bmatrix} 1 & 0 & (D^2 + D + 1)/(D^3 + D^2 + D + 1) \\ 0 & 1 & (D^3 + D + 1)/(D^3 + D^2 + D + 1) \end{bmatrix} \quad (11-73)$$

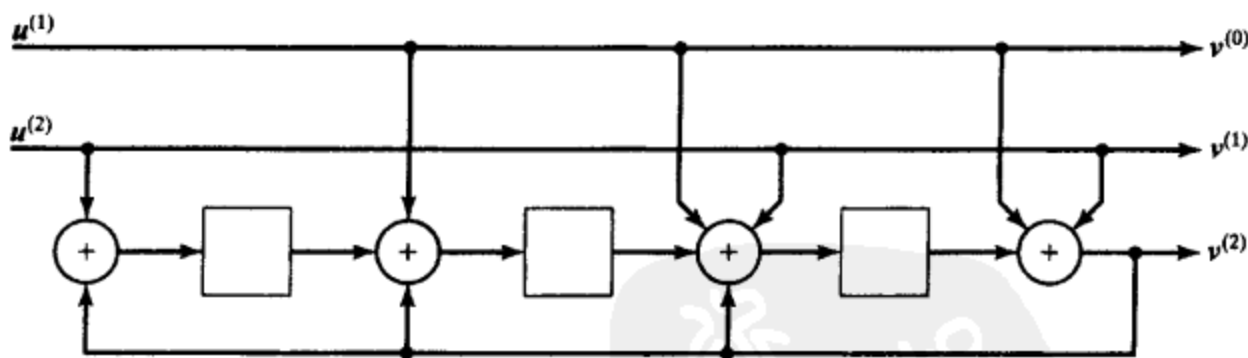
和

$$\mathbf{H}'(D) = \begin{bmatrix} (D^2 + D + 1)/(D^3 + D^2 + D + 1) & (D^3 + D + 1)/(D^3 + D^2 + D + 1) & 1 \end{bmatrix} \quad (11-74)$$

这导致图 11-8b 显示的观测器规范形式系统反馈编码器实现。我们注意到这代表 8 状态(3, 2, 3)编码器, 即观测器规范形式系统反馈编码器实现需要的状态数少于控制器规范形式非系统前馈编码器实现。■



a) 控制器规范形式表示的(3,2,4)非系统前馈编码器



b) 等价的观测器规范形式表示的(3,2,3)系统反馈编码器

图 11-8 用不同规范形式表示的(3, 2, 3)系统反馈编码器

例 11-6 表明了, 对码率 $R = 1/3$ 码, 最小编码器通过控制器规范形式得到。通常, 对由 $1 \times n$ 多项式生成矩阵 $\mathbf{G}(D)$ 或者等价的系统有理函数生成矩阵 $\mathbf{G}'(D)$ 描述的任何码率 $R = 1/n$ 编码器, $\mathbf{G}(D)$ 或者 $\mathbf{G}'(D)$ 的控制器规范形式实现是最小的, 只要保证已经除去 $\mathbf{G}(D)$ 中的所有公因子。(从一个 $1 \times n$ 多项式生成矩阵除去公因子不改变其行空间。)例如, 考虑由如下生成矩阵描述的码率 $R = 1/2$ 非系统前馈编码器

$$\mathbf{G}(D) = [\mathbf{g}^{(0)}(D) \quad \mathbf{g}^{(1)}(D)] = [1 + D^2 \quad 1 + D^3] \quad (11-75)$$

该编码器的控制器规范形式实现有整体约束长度 $v=3$, 即它有 $2^v=8$ 个状态; 但是, 如果我们除去生成多项式 $g^{(0)}(D)$ 和 $g^{(1)}(D)$ 的公因子 $(1+D)$, 我们得到生成矩阵

$$G(D) = [g^{(0)}(D) \quad g^{(1)}(D)] = [1+D \quad 1+D+D^2] \quad (11-76a)$$

或者其系统反馈的等价形式

$$G'(D) = [1 \quad g^{(1)}(D)/g^{(0)}(D)] = [1 \quad (1+D+D^2)/(1+D)] \quad (11-76b)$$

$G(D)$ 和 $G'(D)$ 的控制器规范形式实现都有最小整体约束长度 $v=2$, 即它有 $2^v=4$ 个状态。

类似地, 例 11-7 显示了对码率 $R=2/3$ 码, 最小编码器实现可以用观测器规范形式得到。通常, 对由 $(n-1) \times n$ 多项式生成矩阵 $G(D)$ 和等价的系统有理函数生成矩阵 $G'(D)$ 描述的任何码率 $R=(n-1)/n$ 编码器, $G'(D)$ 的观测器规范形式实现是最小的, 只要保证已经从对应通过 $G'(D)$ 得到的系统有理函数奇偶校验矩阵 $H'(D)$ 的 $1 \times n$ 多项式奇偶校验矩阵 $H(D)$ 中除去所有公因子。(从一个 $1 \times n$ 多项式生成矩阵除去公因子不改变其零空间。)当 $G'(D)$ 中有一个或多个有理函数不可实现时发生例外, 即其分母多项式的延时超过分子多项式的延时, 这里多项式的延时(delay)定义为它的最低次非零项。这样必须找到可能不对应于观测器规范形式的可实现的最小实现。习题 11-9 给出了一个例子。

通常, 由一个 $k \times n$ 生成矩阵 $G(D)$ 产生的码率 $R=k/n$ 卷积码, 也可以通过和 $G(D)$ 具有相同行空间的任意 $k \times n$ 矩阵 $G'(D)$ 产生。如果编码器包含反馈, 则码是递归的; 否则它是非递归的。这些等价生成矩阵中任何一个有最小整体约束长度, 即最小延时元素数的实现都导致最小编码器。总结前述判定最小编码器实现的必要条件以应用于所有实际中通常使用的码率, 即码率 $R=1/n$ 和 $R=(n-1)/n$ 。对 $k>1$ 和 $(n-k)>1$ 的码率, 比如码率 $2/5$, $2/4$ 和 $3/5$, 判定最小编码器实现比较复杂。这些情形中, 最小编码器实现可能需要一种既不是控制器规范形式也不是观测器规范形式的实现。在 Forney^[13] 的开创性文章中首先处理了最小编码器实现的问题。最近, Johannesson 和 Zigangirov^[12] 的书提供了对该问题的全面处理方法。

本书的余下部分中, 当我们提及一个 (n, k, v) 卷积码时, 我们假设它是最小编码器实现, 即 v 是对任意编码器实现的最小整体约束长度。同样地, 当我们使用术语系统(systematic)或非系统(nonsystematic)码时, 我们指由系统或非系统编码器生成的码。最后, 我们注意到, 对控制器规范形式表示的码率 $R=1/n$ 编码器和观测器规范形式表示的码率 $R=(n-1)/n$ 编码器, 整体约束长度 v 等于存储级数 m 。因此, 对码率 $R=1/n$ 和 $R=(n-1)/n$, 我们可以称其为 (n, k, v) 码, 或者一个 (n, k, m) 码。通常, 如果不指定编码器实现的形式, 则 $m \leq v$ 。

到这里我们已经讨论了非系统前馈编码器、系统前馈编码器和系统反馈编码器。编码器的第四种通用形式是非系统反馈编码器(nonsystematic feedback encoders)。

例 11-8 码率 $R=2/3$ 非系统反馈卷积编码器

考虑由如下矩阵生成的码率 $R=2/3$ 非系统递归卷积码(nonsystematic recursive convolutional code, NSRCC)

$$G(D) = \begin{bmatrix} 1/(1+D+D^2) & D/(1+D^3) & 1/(1+D^3) \\ D^2/(1+D^3) & 1/(1+D^3) & 1/(1+D) \end{bmatrix} \quad (11-77)$$

以非系统反馈形式表示。为确定公式(11-77)的控制器(观测器)规范形式实现, $G(D)$ 中每行(列)的有理函数必须具有相同的分母(反馈)多项式, 即我们必须对每行(列)中的有理函数找到最小公分母。这里, 由于 $1/(1+D+D^2) = (1+D)/(1+D^3)$, 以及 $1/(1+D) = (1+D+D^2)/(1+D^3)$, 我们可以将 $G(D)$ 重写成

$$G'(D) = \begin{bmatrix} (1+D)/(1+D^3) & D/(1+D^3) & 1/(1+D^3) \\ D^2/(1+D^3) & 1/(1+D^3) & (1+D+D^2)/(1+D^3) \end{bmatrix} \quad (11-78)$$

现在我们可以直接从 $G'(D)$ 确定由 $G(D)$ 生成的码率 $R = 2/3$ NSRCC 的控制器和观测器规范形式实现。这些实现显示在图 11-9 中。注意到控制器规范形式导致了 64 状态 ($v = 6$) 实现, 而观测器规范形式导致了 512 状态 ($v = 9$) 实现。习题 11-10 表明可以通过把 $G(D)$ 转化成系统反馈形式来得到最小的 8 状态 ($v = 3$) 观测器规范形式实现, 因而导致了 (3, 2, 3) SRCC。

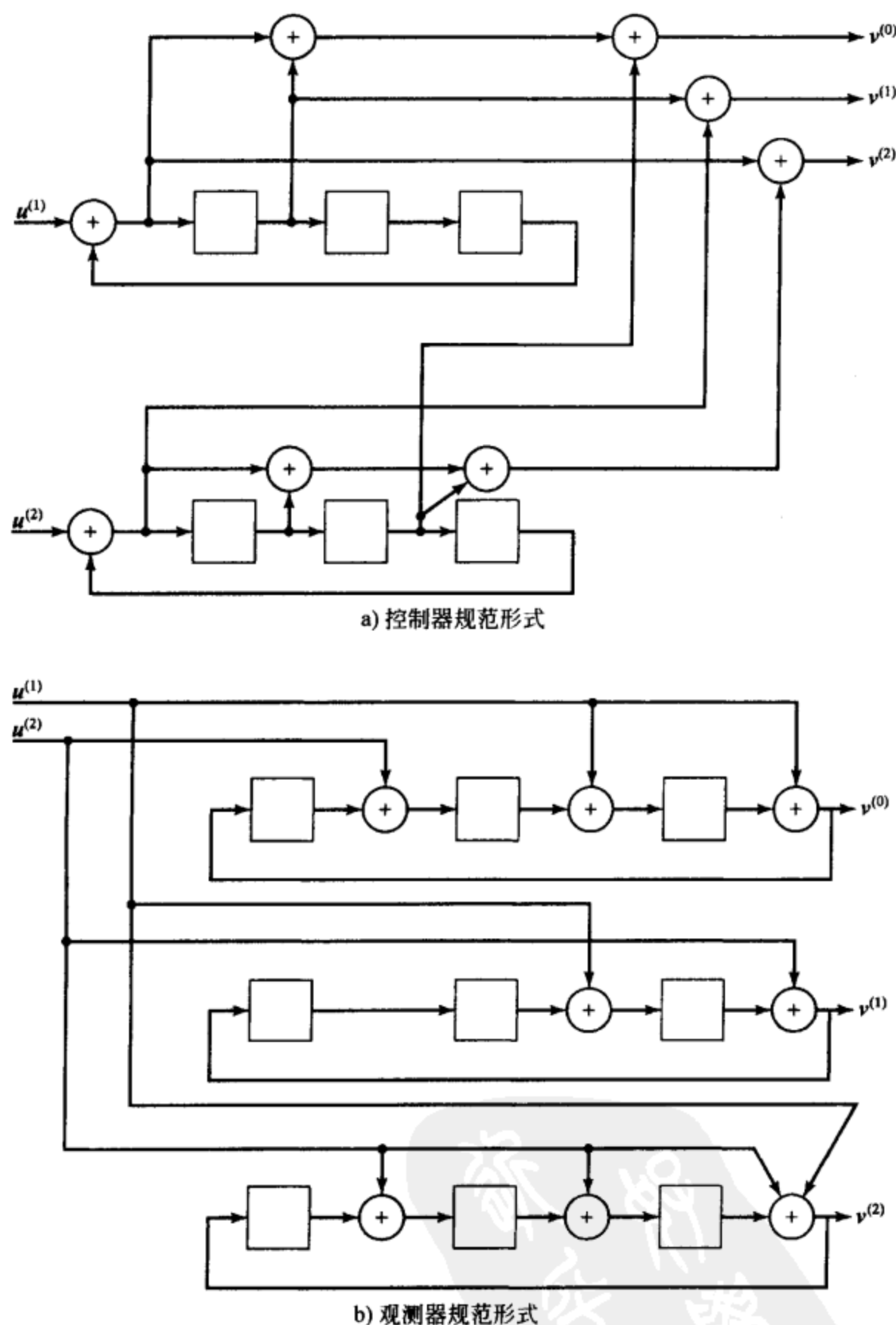


图 11-9 一个非系统反馈编码器的实现

例 11-8 和习题 11-10 说明了, 用于对非系统前馈编码器寻找最小实现的方法可以用到非系统反馈编码器。为总结我们对卷积编码器实现的讨论, 我们在图 11-10 中显示了对如下任意可实现有理函数的控制器和观测器规范形式实现

$$g(D) = f(D)/q(D) = (f_0 + f_1 D + \cdots + f_m D^m) / (1 + q_1 D + \cdots + q_m D^m) \quad (11-79)$$

其中 $q(D)$ 必须有非零常数项以保证可实现性。

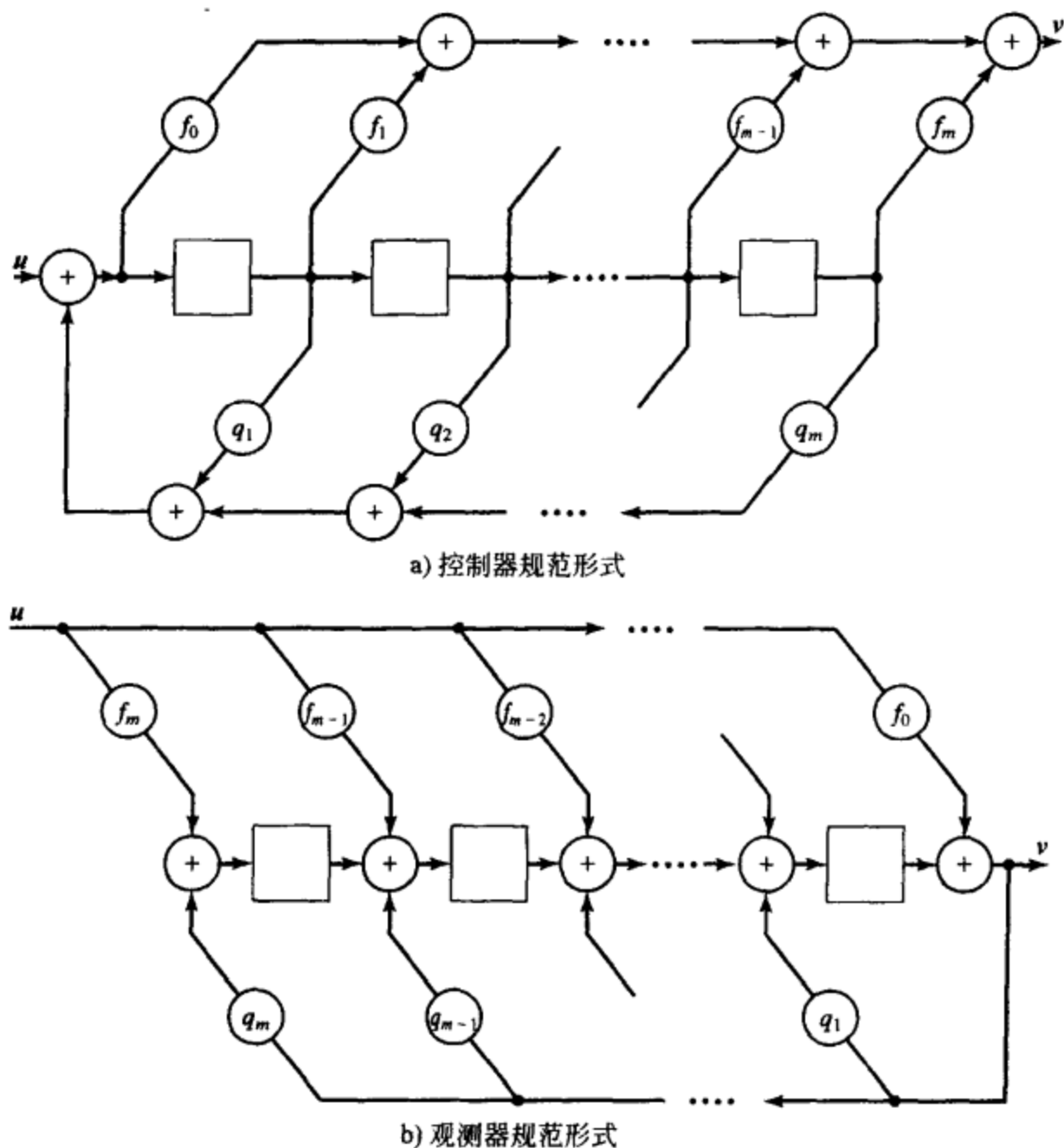


图 11-10 任意可实现有理函数的实现

现在我们可以指定用于在本书余下部分中描述实际中重要码率，即码率 $R = 1/n$ 和 $R = (n-1)/n$ 的卷积码的记号了。任意码率 $R = 1/n$ 卷积码可以通过最小非系统前馈编码器描述(见习题 11-6)，其多项式生成矩阵

$$G(D) = [g^{(0)}(D) \quad g^{(1)}(D) \quad \cdots \quad g^{(n-1)}(D)] \quad (11-80a)$$

或者其有理函数系统反馈等价形式

$$G'(D) = [1 \quad g^{(1)}(D)/g^{(0)}(D) \quad \cdots \quad g^{(n-1)}(D)/g^{(0)}(D)] \quad (11-80b)$$

(如果 $g^{(0)}(D) = 1$ ，公式(11-80)描述了系统前馈编码器。)通过指定 n 个生成多项式 $g^{(0)}(D)$, $g^{(1)}(D)$, $\cdots$, $g^{(n-1)}(D)$ ，或者 n 个等价的生成序列 $g^{(0)}$, $g^{(1)}$, $\cdots$, $g^{(n-1)}$ 描述一个码。类似地，任何码率 $R = (n-1)/n$ 卷积码可以通过最小系统反馈编码器描述(见习题 11-7)，其有理函数奇偶校验矩阵为

$$H'(D) = [h^{(0)}(D)/h^{(n-1)}(D) \quad \cdots \quad h^{(n-2)}(D)/h^{(n-1)}(D) \quad 1] \quad (11-81a)$$

或者其多项式非系统前馈等价形式

$$H(D) = [h^{(0)}(D) \quad \cdots \quad h^{(n-2)}(D) \quad h^{(n-1)}(D)] \quad (11-81b)$$

(如果 $h^{(n-1)}(D) = 1$ ，公式(11-81)描述了一个系统前馈编码器。)通过指定 n 个奇偶校验多项式 $h^{(0)}(D)$, $h^{(1)}(D)$, $\cdots$, $h^{(n-1)}(D)$ ，或者 n 个等价的奇偶校验序列 $h^{(0)}$, $h^{(1)}$, $\cdots$,

$h^{(n-1)}$ 描述一个码。

在公式(11-81)的记号中, 对码率 $R = (n-1)/n$ 码, 输出校验序列 $v^{(n-1)}$ 和反馈奇偶校验序列 $h^{(n-1)}$ 有关, $(n-1)$ 个输出信息序列 $v^{(0)}, \dots, v^{(n-2)}$ (即输入信息序列 $u^{(1)}, \dots, u^{(n-1)}$) 和奇偶校验序列 $h^{(0)}, \dots, h^{(n-2)}$ 有关。同样地, 在编码器的观测器规范形式实现中, 输出校验序列 $v^{(n-1)}$ 在图的底部, $(n-1)$ 个输出信息序列, 按照 $v^{(0)}, \dots, v^{(n-2)}$ 的次序, 在顶部。(例如, 参考图 11-8b 中输出序列的标号)。由于涉及在第 18 章讨论的仅使用码率 $R = (n-1)/n$ 码的网格编码调制系统设计的原因, 在文献中已经习惯于颠倒输入和输出序列的标号次序。换言之, 不同于公式(11-81)中的标号, 描述码率 $R = (n-1)/n$ 码的奇偶校验矩阵通常写成

$$H'(D) = [h^{(n-1)}(D)/h^{(0)}(D) \quad \dots \quad h^{(1)}(D)/h^{(0)}(D) \quad 1] \quad (11-82a)$$

和

$$H(D) = [h^{(n-1)}(D) \quad \dots \quad h^{(1)}(D) \quad h^{(0)}(D)] \quad (11-82b)$$

(对系统前馈编码器, 新标号意味着 $h^{(0)}(D) = 1$ 和 $h^{(j)}(D) = g_{n-j}^{(n-1)}(D)$, $j = 1, 2, \dots, n-1$ 。)这里分别地, 输出奇偶序列 $v^{(0)}$ 和反馈奇偶校验序列 $h^{(0)}$ 有关, $(n-1)$ 个输出信息序列 $v^{(1)}, \dots, v^{(n-1)}$ (即输入信息序列 $u^{(1)}, \dots, u^{(n-1)}$) 和奇偶校验序列 $h^{(1)}, \dots, h^{(n-1)}$ 有关。为对应这种记号, 编码器的观测器规范形式实现被重新标号, 使输出奇偶序列 $v^{(0)}$ 在图的底部, $(n-1)$ 输出信息序列, 按照 $v^{(n-1)}, \dots, v^{(0)}$ 的次序在顶部, $(n-1)$ 个输入序列也相应地重新进行标记。作为一个例子, 图 11-8b 中系统反馈编码器的 (3, 2, 3) 观测器规范形式实现的重新标记后示于图 11-11 中。为避免和该领域已出版的文献混

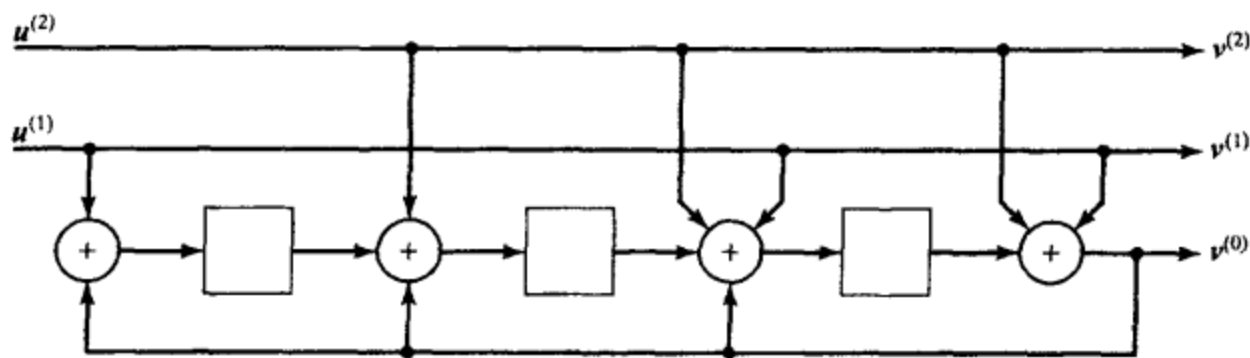


图 11-11 图 11-8b 中 (3, 2, 3) 系统反馈编码器重新标号后的形式

淆, 在本书的余下部分中, 我们都使用公式(11-82)的修正记号和图 11-11 的重新标号来描述码率 $R = (n-1)/n$ 码。[⊖]

有些情形下, 希望不通过译码直接从码字中恢复带噪声的信息序列。例如, 当正在测试信道噪声条件, 当译码器出故障了, 或者当信道 SNR 足够高不需要译码时, 情况可能是这样。通常, 这种处理需要一个变换器来从码字恢复信息序列。对系统编码器来说变换器是没有必要的, 因为可以直接获得信息序列。另一方面, 对非系统编码器, 必须存在一个 $n \times k$ 逆矩阵 $G^{-1}(D)$ 满足

$$G(D)G^{-1}(D) = ID^l \quad (11-83)$$

对某个 $l \geq 0$ 成立, 其中 I 是 $k \times k$ 单位矩阵。则公式(11-32)和公式(11-83)意味着

$$V(D)G^{-1}(D) = U(D)G(D)G^{-1}(D) = U(D)D^l \quad (11-84)$$

信息序列可以使用 l 时间单元延迟, 从令 $V(D)$ 为 n 输入、 k 输出线性时序电路的输入得到的码字中恢复, 该电路的传递函数矩阵是 $G^{-1}(D)$ 。

对一个 $(n, 1, v)$ 卷积编码器, 生成矩阵 $G(D)$ 有延时 l 的前馈逆 $G^{-1}(D)$, 当且仅当

$$\text{GCD}[g^{(0)}(D), g^{(1)}(D), \dots, g^{(n-1)}(D)] = D^l \quad (11-85)$$

⊖ 除了在 13.5 ~ 13.7 节和 21 章中, 为和系统前馈编码器方面的文献一致, 使用图 11-5 的标号来描述码率 $R = (n-1)/n$ 码。

对某个 $l \geq 0$ 成立, 其中 GCD 代表最大公因子。对一个 $k > 1$ 的 (n, k, v) 编码器, 令 $\Delta_i(D)$, $i = 1, 2, \dots, \binom{n}{k}$ 是生成矩阵 $G(D)$ 的 $\binom{n}{k}$ 个不同的 $k \times k$ 子矩阵的行列式。则延时 l 的前馈逆存在当且仅当

$$\text{GCD}[\Delta_i(D) : i = 1, 2, \dots, \binom{n}{k}] = D^l \quad (11-86)$$

对某个 $l \geq 0$ 成立。Massey 和 Sain [14] 给出了构造前馈逆 $G^{-1}(D)$ 的方法。 ■

例 11-9 前馈编码器逆

1) 对图 11-11 的 $(2, 1, 3)$ 编码器, 生成矩阵由下式给出

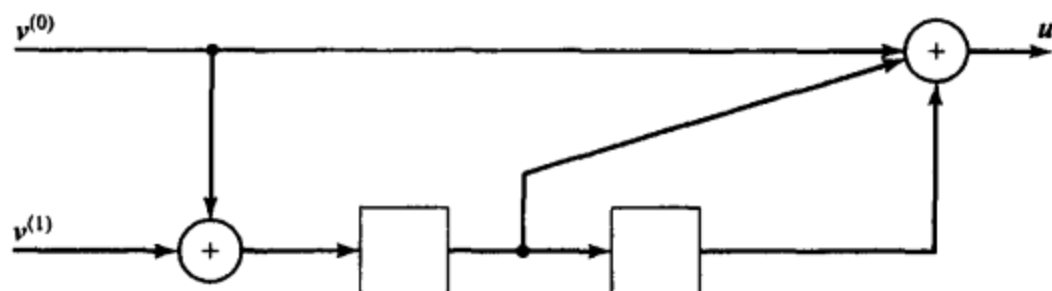
$$G(D) = [g^{(0)}(D) \quad g^{(1)}(D)] = [1 + D^2 + D^3 \quad 1 + D + D^2 + D^3] \quad (11-87a)$$

$$\text{GCD}[g^{(0)}(D), g^{(1)}(D)] = \text{GCD}[1 + D^2 + D^3, 1 + D + D^2 + D^3] = 1 \quad (11-87b)$$

以及传递函数矩阵

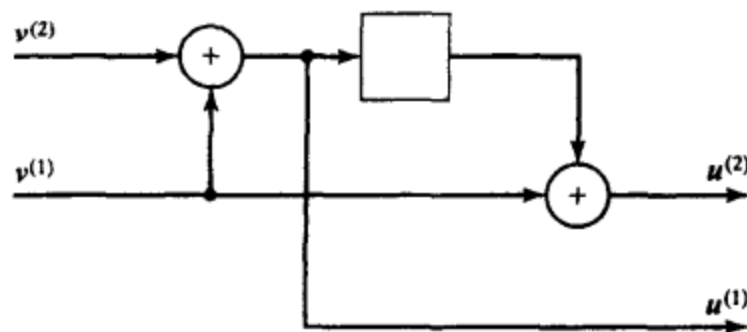
$$G^{-1}(D) = \begin{bmatrix} 1 + D + D^2 \\ D + D^2 \end{bmatrix} \quad (11-88)$$

提供了延时为 0 的前馈逆, 即 $G(D)G^{-1}(D) = 1$ 。该编码器逆的一种实现示于图 11-12a 中。



a) $(3,1,2)$ 编码器的前馈编码器逆

2) 对图 11-12b 显示的 $(3, 2, 2)$ 编码器, 其生成矩阵 $G(D)$ 由公式 (11-34) 给出, $G(D)$ 的 2×2 子矩阵产生行列式 $1 + D + D^2$, $1 + D^2$ 和 1。由于 $\text{GCD}[1 + D + D^2, 1 + D^2, 1] = 1$ (11-89)



b) $(3,2,2)$ 编码器的前馈编码器逆

图 11-12 前馈编码器逆的实现

所以存在延时为 0 的前馈逆。所求的传递函数矩阵由下式给出

$$G^{-1}(D) = \begin{bmatrix} 0 & 0 \\ 1 & 1 + D \\ 1 & D \end{bmatrix} \quad (11-90)$$

其实现显示在图 11-12b 中。 ■

为理解当一个前馈逆不存在时将发生什么, 最好考虑另一个例子。 ■

例 11-10 恶性编码器

考虑一个 $(2, 1, 2)$ 编码器, 其生成矩阵由下式给出

$$G(D) = [g^{(0)}(D) \quad g^{(1)}(D)] = [1 + D \quad 1 + D^2] \quad (11-91a)$$

这里我们看到

$$\text{GCD}[g^{(0)}(D), g^{(1)}(D)] = \text{GCD}[1 + D, 1 + D^2] = 1 + D \quad (11-91b)$$

因而前馈逆不存在。如果信息序列是 $u(D) = \frac{1}{1+D} = 1 + D + D^2 + \dots$, 则输出序列是 $v^{(0)}(D) = 1$ 和 $v^{(1)}(D) = 1 + D$, 即码字仅有重量 3, 即使信息序列有无限重量。如果该码字在 BSC 上传输, 其中三个非零比特被信道噪声变为零, 则接收序列是全零。最大似然译码器则会产生全零码字

作为其估计,因为这是个有效码字,并且正好和接收序列相同。因此,估计的信息序列是 $u(D) = 0(D)$, 意味着由有限数量(这里仅有三个)的信道错误引起了无限数量的译码错误。明显地,这是十分不希望的,译码器被称为遭受了恶性误码扩散,即它是一个恶性译码器(catastrophic encoder)。

因此,在选择编码器用于差错控制编码系统时,避免选择恶性编码器是非常重要的。Massey 和 Sain^[14]已经指出一个编码器是非恶性的充分必要条件是公式(11-85)和(11-86)。因此,任意前馈逆存在的编码器是非恶性的。因此断定系统编码器总是非恶性的,因为存在一个平凡的前馈逆。最小非系统编码器也是非恶性的,但是非最小编码器可以是恶性的^[12]。注意到,比如公式(11-75)的非最小编码器不满足公式(11-85),因此它是恶性的。本书后面第 12、13、16、18 和 21 章给出的好卷积编码器的表格只列出了非恶性编码器。

一个 (n, k, v) 卷积编码器可以产生无限长编码序列(码字)。由于每 k 个信息比特产生 n 个编码比特, $R = k/n$ 称为卷积编码器的码率。对 h 个时间单元的有限长度信息序列,或者说是 $K^* = kh$ 比特,习惯从全零状态开始编码,即移位寄存器的初始内容是全零。同样地,为便于译码,在最后的分组进入编码器之后,有必要使编码器回到全零状态,这一点将在第 12 章和第 13 章中看出。这要求额外的 m 个输入分组进入编码器迫使它回到全零状态。对前馈编码器,可以从编码器图看出,这 m 个收尾分组必须是全零,而对反馈编码器,收尾分组依赖于信息序列(见习题 11-12)。这些收尾分组不是信息序列的一部分,因为它们固定的,不能任意选择。(我们注意到,这里对 $v < km$ 的控制器规范形式编码器实现,仅需要 v 个收尾比特。因此, m 个收尾分组中的 $km - v$ 比特可以是信息比特。例如,对图 11-3 所示的 $(4, 3, 3)$ 编码器实现,仅需要 $v = 3$ 个收尾比特,因此 2 个收尾分组中余下的 3 个比特可以是信息比特。相同的结论对观测器规范形式实现也成立(见习题 11-13)。然而为了简化,我们在接下来的推导中忽略这些特殊情形。)由收尾导致的相应码字具有长度 $N = n(h + m)$, 包含 m 个输出分组,或者说是 nm 个输出比特。这样,收尾卷积码可以看成是一个有 $K^* \times n$ 维生成阵 G 的 $(N, K^*) = [n(h + m), kh]$ 线性分组码。分组(收尾的卷积)码码率由 $R_t = \frac{K^*}{N} = \frac{kh}{n(h + m)}$ 给出,信息比特的数量和码字长度的比值。如果 $h \gg m$, 则 $h/(h + m) \approx 1$, 分组(收尾的卷积)码码率 R_t 和卷积编码器码率 R 近似相等。这代表卷积码工作的通常模式,我们一般不再区分卷积编码器码率 R 和对应的分组(收尾的卷积)码的码率 R_t , 即假定 $R = R_t$ 。如果 h 很小,则作为分组(收尾的卷积)码码率 R_t 的比值 $\frac{kh}{n(h + m)}$ 比卷积编码器码率 R 减少了一个分数

$$\frac{\frac{k}{n} - \frac{kh}{n(h + m)}}{k/n} = \frac{m}{h + m} \quad (11-92)$$

称为码率损失系数(fractional rate loss)。例如,对图 11-11 中的 $(2, 1, 3)$ 编码器和信息序列, $h = 5$, 码率损失系数是 $\frac{3}{8} = 37.5\%$, 但对长度 $h = 1000$ 的信息序列,码率损失系数仅仅是 $\frac{3}{1003} = 0.3\%$ 。

对短信息序列,码率损失系数可以通过简单地发送 nm 个收尾比特并且考虑最初的 $N^* = nh$ 个编码器输出比特作为码字而得到消除,但是在第 12 和第 13 章中将会看到,如果不发送收尾比特,码的性能将会降低。克服这个问题的一种方法是允许编码器从任意状态开始——即移位寄存器的最初内容没有必要是全零——并且在信息序列的末尾迫使它回到同一

个状态。如果适当选择起始状态,则不需要额外的比特来迫使编码器回到同一个状态,因而不会引发码率损失系数。这种技术导致了所谓的咬尾码,这将在第12.7节中更详细地讨论。

11.2 卷积码的结构特点

由于卷积编码器是线性时序电路,其运算可以用状态图描述。编码器的状态定义为其移位寄存器的内容。对一个控制器规范形式的 (n, k, v) 编码器,第 i 个移位寄存器在时间单元 l (当 $u_l^{(1)}, u_l^{(2)}, \dots, u_l^{(k)}$ 是编码器输入)包含 v_i 个比特,记为 $s_{l-1}^{(i)}, s_{l-2}^{(i)}, \dots, s_{l-v_i}^{(i)}$,其中 $s_{l-1}^{(i)}$ 代表最左边延时单元的内容, $s_{l-v_i}^{(i)}$ 代表最右边单元的内容, $1 \leq i \leq k$ 。

定义 11-6 在时间单元 l 的编码器状态(encoder state) σ_l 是二进制 v 维向量

$$\sigma_l = (s_{l-1}^{(1)} s_{l-2}^{(1)} \dots s_{l-v_1}^{(1)} s_{l-1}^{(2)} s_{l-2}^{(2)} \dots s_{l-v_2}^{(2)} \dots s_{l-1}^{(k)} s_{l-2}^{(k)} \dots s_{l-v_k}^{(k)}) \quad (11-93)$$

因此,从公式(11-20)我们看出总共存在 2^v 个不同的可能状态。对 $(n, 1, v)$ 编码器,在时间单元 l 的编码器状态简单地就是

$$\sigma_l = (s_{l-1} \quad s_{l-2} \quad \dots \quad s_{l-v}) \quad (11-94)$$

对前馈编码器的情形,我们从编码器图(例如,见图11-1)看出,第 i 个移位寄存器在时间单元 l 的内容是它前面的 v_i 个输入,即 $u_{l-1}^{(i)}, u_{l-2}^{(i)}, \dots, u_{l-v_i}^{(i)}$,因此在时间单元 l 的编码器状态由下式给出

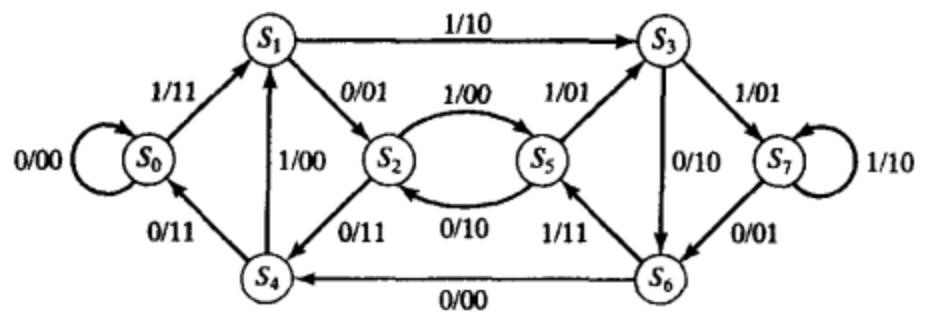
$$\sigma_k = (u_{l-1}^{(1)} u_{l-2}^{(1)} \dots u_{l-v_1}^{(1)} u_{l-1}^{(2)} u_{l-2}^{(2)} \dots u_{l-v_2}^{(2)} \dots u_{l-1}^{(k)} u_{l-2}^{(k)} \dots u_{l-v_k}^{(k)}) \quad (11-95)$$

对一个 $(n, 1, v)$ 编码器,公式(11-93)简单地成为

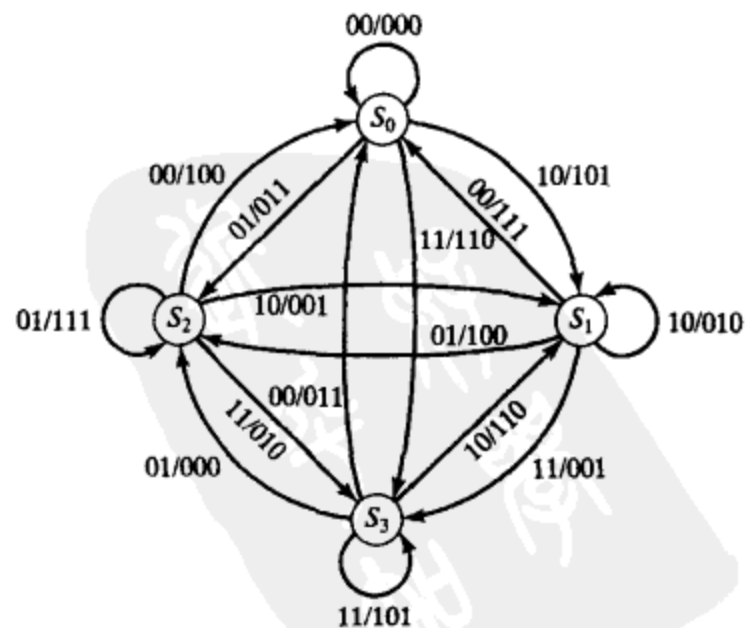
$$\sigma = (u_{l-1} \quad u_{l-2} \quad \dots \quad u_{l-v}) \quad (11-96)$$

对观测器规范形式编码器实现,在时间单元 l 的编码器状态 σ_l 也可以定义成公式(11-93),只是这种情形下,通常存在总共为 n 个而不是 k 个移位寄存器。同样,观测器规范形式的前馈编码器不具备移位寄存器内容为其 v_i 个以前输入的性质,即公式(11-95)和(11-96)只适用于控制器规范形式的前馈编码器实现。

每个新输入的 k 比特分组促使编码器移位,即转移到一个新的状态。因此,在状态图中有 2^k 个分支离开每个状态,每个对应于一个不同的输入分组。从而对一个 $(n, 1, v)$ 编码器,仅有两个分支离开每个状态。状态图中的每个状态标有促使其转移的 k 个输入($u_l^{(1)}, u_l^{(2)}, \dots, u_l^{(k)}$)和 n 个相应的输出($v_l^{(0)}, v_l^{(1)}, \dots, v_l^{(n-1)}$)。图11-1中 $(2, 1, 3)$ 编码器和图11-2中 $(3, 2, 2)$ 编码器的前馈控制器规范形式实现的状态图分别显示在图11-13a和图11-13b中。状态标为 $S_0, S_1, \dots, S_{2^v-1}$,其中习惯上 S_i 代表其二进制 v 维向量表示 $b_0, b_1, \dots, b_{v-1}$ 等价于整数 $i = b_0 2^0 + b_1 2^1 + \dots + b_{v-1} 2^{v-1}$ 的状态。例如,对图11-1中的 $(2, 1, 3)$ 编码器,如果在时间单元 l 移位寄存器的内



a) 图11-1中的 $(2,1,3)$ 编码器状态图



b) 图11-2中的 $(3,2,2)$ 编码器的编码器状态图

图11-13 编码器状态图

容由 $\sigma_l = (s_{l-1}s_{l-2}s_{l-3}) = (u_{l-1}u_{l-2}u_{l-3})$ 给出, 则在时间单元 l 的状态记为 S_i , 其中 $i = u_{l-1} + 2u_{l-2} + 4u_{l-3}$ 。

这里我们注意到, 对时不变卷积编码器 (time-invariant convolutional encoder), 即生成矩阵和相应的编码器图不随时间而改变的编码器, 编码器状态图也是时不变的。因此同一个状态图 (比如, 见图 11-13 中的那些) 是对所有的时间单元 l 有效的。本书不直接处理时变卷积编码器 (time-varying convolutional encoder), 但我们指出将在第 16 章讲述的 turbo 码类是由一种特殊的时变卷积编码器生成的。

假定编码器开始在状态 S_0 (全零状态), 通过在状态图中沿着由信息序列决定的路径且记录分支上标记的相应输出, 我们可以得到对应于任何给定信息序列的码字。沿着最后的信息分组, 收尾的编码器通过一个附加到信息序列的 m 个输入分组序列回到状态 S_0 。例如, 在图 11-13a 中, 如果 $u = (11101)$, 编码通过附加 $m = 3$ 个零到 u 上收尾, 其码字由 $v = (11, 10, 01, 01, 11, 10, 11, 11)$ 给出。

现在我们暂时回到上一节引入的恶性编码器问题上。一个编码器是恶性的, 当且仅当状态图除了有一个围绕状态 S_0 的零重量圈之外, 还包含一个零输出重量圈。例 11-10 的恶性编码器状态图显示在图 11-14 中。注意到, 围绕状态 S_3 的单个分支圈有零输出重量。

可以修改状态图来对所有非零码字 (codeword) 的汉明重量给出一种完整描述, 即码的码重枚举函数 (codeword weight enumerating function, WEF)。我们从考虑不收尾的编码器开始, 即信息序列可以是任意长度。状态 S_0 拆成初始状态和终止状态, 删除了围绕状态 S_0 的零重量分支, 每个分支都标以分支增益 (branch gain) X^d , 其中 d 是该分支上 n 个编码比特的重量。每条连接初始状态和终止状态的路径, 代表一个分叉于且再合并于状态 S_0 恰好一次的非零码字。(我们不计数从来不再合并于状态 S_0 的码字, 因为对非恶性编码器, 它们必须具有无限重量。同样也不计数分叉且再合并于状态 S_0 多于一次的码字, 因为它们可以看成更短码字的序列。) 路径增益 (path gain) 是沿一条路径上的分支增益的乘积, 相应码字重量是路径增益中 X 的幂。图 11-1 和 11-2 中编码器的修正状态图分别显示在图 11-15a 和 11-15b 中。

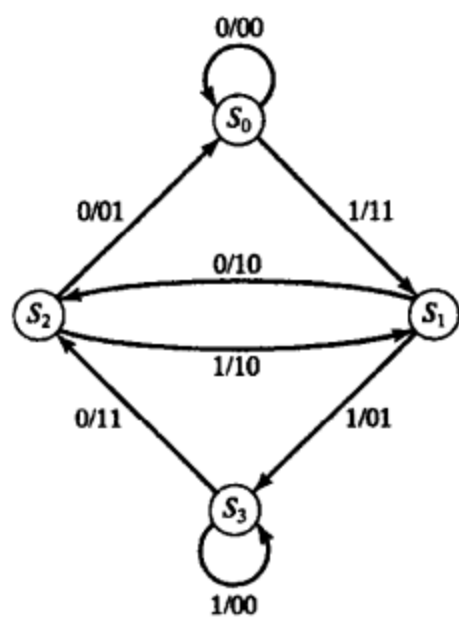
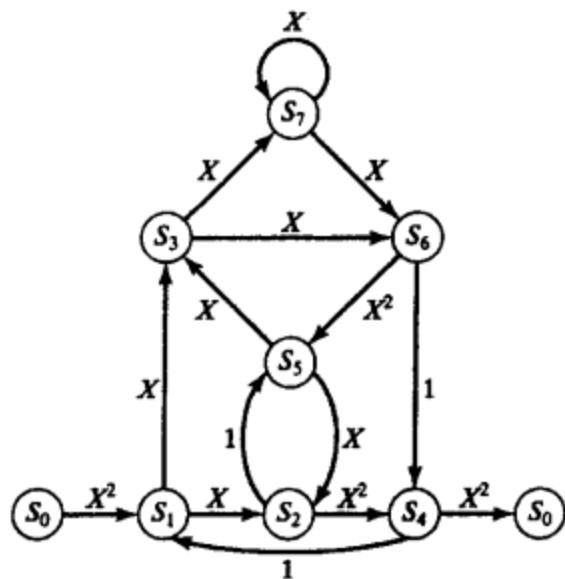
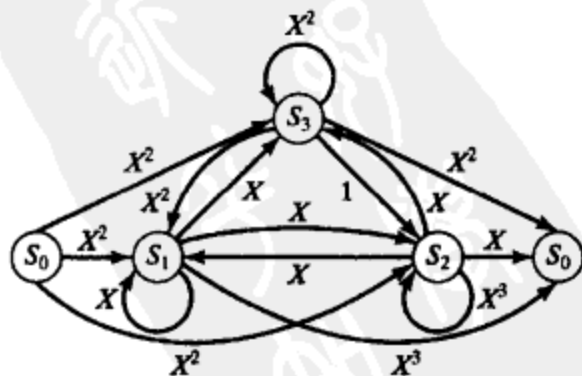


图 11-14 (2, 1, 2) 恶性编码器的状态图



a) 图 11-1 中 (2, 1, 3) 编码器的修正编码器状态图



b) 图 11-2 中 (3, 2, 2) 编码器的修正编码器状态图

图 11-15 修正编码器状态图

例 11-11 计算状态图中的路径增益

a. 在图 11-15a 中代表状态序列 $S_0S_1S_3S_7S_9S_5S_2S_4S_0$ 的路径有路径增益 $X^2 \cdot X^1 \cdot X^1 \cdot X^1 \cdot X^2 \cdot X^1 \cdot X^2 \cdot X^2 = X^{12}$, 相应的码字有重量 12。

b. 在图 11-15b 中代表状态序列 $S_0S_1S_3S_2S_0$ 的路径有路径增益 $X^2 \cdot X^1 \cdot X^0 \cdot X^1 = X^4$, 相应的码字有重量 4。

通过把编码器的修正状态图看成是信号流图 (signal flow graph), 并使用梅森 (Mason) 增益公式^[15]来计算其传递函数 (transfer function), 我们可以计算码的码字 WEF:

$$A(X) = \sum_d A_d X^d \quad (11-97)$$

其中 A_d 是重量为 d 的码字数目。在信号流图中, 连接初始状态和终止状态, 且任意状态不经过两次的路径称为前向路径 (forward path)。令 F_i 是第 i 条前向路径的增益。从任何状态出发又回到该状态, 且任何其他状态不经过两次的闭路称为环路 (cycle)。令 C_i 是第 i 个环路的增益。在一组环路中, 若没有一个状态是属于此集合的一个以上的环路时, 就称之为不相连 (nontouching)。令 $\{i\}$ 是所有环路的集合, $\{i', j'\}$ 是所有不相连环路对集合, $\{i'', j'', l''\}$ 是所有三重不相连的环路集合等。我们定义

$$\Delta = 1 - \sum_i C_i + \sum_{i', j'} C_{i'} C_{j'} - \sum_{i', j', l'} C_{i'} C_{j'} C_{l'} + \dots \quad (11-98)$$

其中 $\sum_i C_i$ 是环路增益的和, $\sum_{i', j'} C_{i'} C_{j'}$ 是两个不相连环路的环路增益乘积在所有不相连环路对上求和, $\sum_{i', j', l'} C_{i'} C_{j'} C_{l'}$ 是三个不相连环路的环路增益乘积在所有不相交的三重上求和等。最后, Δ_i 定义和 Δ 完全一致, 不同之处在于图的分割不涉及第 i 个前向路径, 即当计算 Δ_i 时, 沿着第 i 条前向路径上的所有状态, 以及连接到那些状态的所有分支都从图中删除了。我们现在可以写出梅森公式来计算一个图的传递函数 $A(X)$ 如下

$$A(X) = \frac{\sum_i F_i \Delta_i}{\Delta} \quad (11-99)$$

其中分子中的求和遍历所有前向路径。我们给出一些例子来说明这个过程。

例 11-12 计算 (2, 1, 3) 码的 WEF

考虑图 11-1 中 (2, 1, 3) 非系统前馈编码器生成的码, 其修正状态图示于图 11-15a 中。图 11-15a 中的图有 11 个环路:

环路 1:	$S_1S_3S_7S_6S_5S_2S_4S_1$	$(C_1 = X^8)$
环路 2:	$S_1S_3S_7S_6S_4S_1$	$(C_2 = X^3)$
环路 3:	$S_1S_3S_6S_5S_2S_4S_1$	$(C_3 = X^7)$
环路 4:	$S_1S_3S_6S_4S_1$	$(C_4 = X^2)$
环路 5:	$S_1S_2S_5S_3S_7S_6S_4S_1$	$(C_5 = X^4)$
环路 6:	$S_1S_2S_5S_3S_6S_4S_1$	$(C_6 = X^3)$
环路 7:	$S_1S_2S_4S_1$	$(C_7 = X^3)$
环路 8:	$S_2S_5S_2$	$(C_8 = X)$
环路 9:	$S_3S_7S_6S_5S_3$	$(C_9 = X^5)$
环路 10:	$S_3S_6S_5S_3$	$(C_{10} = X^4)$
环路 11:	S_7S_7	$(C_{11} = X)$

有 10 个不相连环路对:

环路对 1:	(环路 2, 环路 8)	$(C_2 C_8 = X^4)$
环路对 2:	(环路 3, 环路 11)	$(C_3 C_{11} = X^8)$
环路对 3:	(环路 4, 环路 8)	$(C_4 C_8 = X^3)$
环路对 4:	(环路 4, 环路 11)	$(C_4 C_{11} = X^3)$
环路对 5:	(环路 6, 环路 11)	$(C_6 C_{11} = X^4)$
环路对 6:	(环路 7, 环路 9)	$(C_7 C_9 = X^8)$
环路对 7:	(环路 7, 环路 10)	$(C_7 C_{10} = X^7)$
环路对 8:	(环路 7, 环路 11)	$(C_7 C_{11} = X^4)$
环路对 9:	(环路 8, 环路 11)	$(C_8 C_{11} = X^2)$
环路对 10:	(环路 10, 环路 11)	$(C_{10} C_{11} = X^5)$

有 2 条三重不相连环路:

三重环路 1: (环路 4, 环路 8, 环路 11) $(C_4 C_8 C_{11} = X^4)$

三重环路 2: (环路 7, 环路 10, 环路 11) $(C_7 C_{10} C_{11} = X^8)$

没有其他不相连环路集。因此

$$\begin{aligned}\Delta &= 1 - (X^8 + X^3 + X^7 + X^2 + X^4 + X^3 + X^3 + X + X^5 + X^4 + X) \\ &\quad + (X^4 + X^8 + X^3 + X^3 + X^4 + X^8 + X^7 + X^4 + X^2 + X^5) - (X^4 + X^8) \\ &= 1 - 2X - X^3\end{aligned}\quad (11-100)$$

图 11-15a 的图中有 7 条前向路径:

前向路径 1: $S_0 S_1 S_3 S_7 S_6 S_5 S_2 S_4 S_0$ $(F_1 = X^{12})$

前向路径 2: $S_0 S_1 S_3 S_7 S_6 S_4 S_0$ $(F_2 = X^7)$

前向路径 3: $S_0 S_1 S_3 S_6 S_5 S_2 S_4 S_0$ $(F_3 = X^{11})$

前向路径 4: $S_0 S_1 S_3 S_6 S_4 S_0$ $(F_4 = X^6)$

前向路径 5: $S_0 S_1 S_2 S_5 S_3 S_7 S_6 S_4 S_0$ $(F_5 = X^8)$

前向路径 6: $S_0 S_1 S_2 S_5 S_3 S_6 S_4 S_0$ $(F_6 = X^7)$

前向路径 7: $S_0 S_1 S_2 S_4 S_0$ $(F_7 = X^7)$

前向路径 1 和 5 涉及图中所有状态, 因此不接触这些路径的子图不包含状态。因此

$$\Delta_1 = \Delta_5 = 1 \quad (11-101)$$

不接触前向路径 3 和 6 的子图示于图 11-16a, 因此

$$\Delta_3 = \Delta_6 = 1 - X \quad (11-102)$$

不接触前向路径 2 的子图示于图 11-16b, 因此

$$\Delta_2 = 1 - X \quad (11-103)$$

不接触前向路径 4 的子图示于图 11-16c, 因此

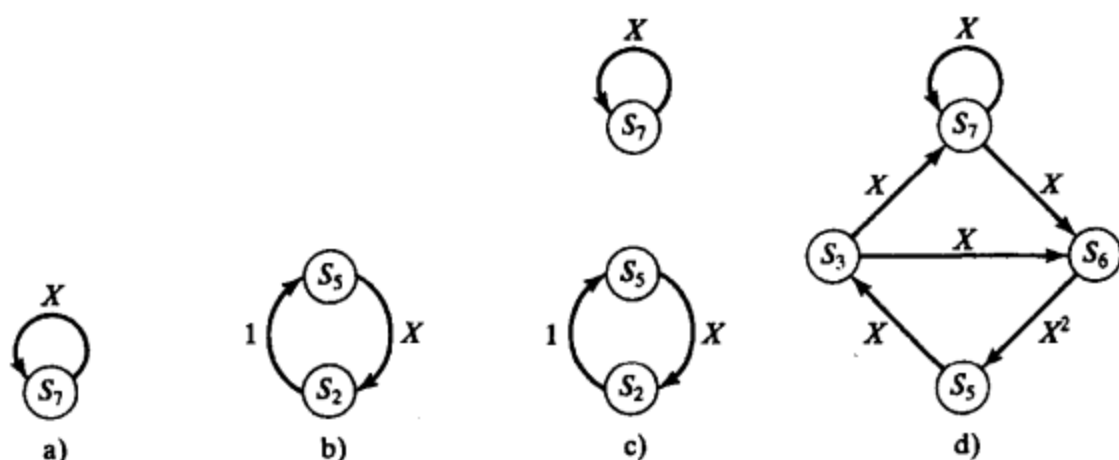
$$\Delta_4 = 1 - (X + X) + (X^2) = 1 - 2X + X^2 \quad (11-104)$$

不接触前向路径 7 的子图示于图 11-16d, 因此

$$\Delta_7 = 1 - (X + X^4 + X^5) + (X^5) = 1 - X - X^4 \quad (11-105)$$

该图的传递函数, 即对应(2, 1, 3)码的 WEF 由下式给出

$$\begin{aligned}A(X) &= \frac{X^{12} \cdot 1 + X^7(1 - X) + X^{11}(1 - X) + X^6(1 - 2X + X^2) + X^8 \cdot 1 + X^7(1 - X) + X^7(1 - X - X^4)}{1 - 2X - X^3} = \frac{X^6 + X^7 - X^8}{1 - 2X - X^3} \\ &= X^6 + 3X^7 + 5X^8 + 11X^9 + 25X^{10} + \dots\end{aligned}\quad (11-106)$$

图 11-16 例 11-12 中计算 Δ_i 的子图

码字 WEF $A(X)$ 提供了对分叉于且再合并于状态 S_0 恰好一次的所有非零码字重量分布的完整描述。这里，有一个重量为 6 的码字，三个重量为 7，五个重量为 8 等。

例 11-13 计算 (3, 2, 2) 码的 WEF

考虑图 11-2 中 (3, 2, 2) 非系统前馈编码器生成的码，其修正状态图示于图 11-15b 中。可以看出(见习题 11-17)，图 11-15b 的图中有 8 个环路，6 对不相连环路和 1 个三重不相连环路，则

$$\begin{aligned} \Delta &= 1 - (X^2 + X^4 + X^3 + X + X^2 + X + X^2 + X^3) + (X^6 + X^2 + X^4 + X^3 + X^4 + X^5) - (X^6) \\ &= 1 - 2X - 2X^2 - X^3 + X^4 + X^5 \end{aligned} \quad (11-107)$$

图 11-15b 的图中有 15 条前向路径(见习题 11-8)，则

$$\begin{aligned} \sum_i F_i \Delta_i &= X^5(1 - X - X^2 - X^3 + X^5) + X^4(1 - X^2) + X^6 \cdot 1 + X^5(1 - X^3) + X^4 \cdot 1 \\ &\quad + X^3(1 - X - X^2) + X^6(1 - X^2) + X^6 \cdot 1 + X^5(1 - X) + X^8 \cdot 1 \\ &\quad + X^4(1 - X - X^2 - X^3 + X^4) + X^7(1 - X^3) + X^6 \cdot 1 + X^3(1 - X) + X^6 \cdot 1 \\ &= 2X^3 + X^4 + X^5 + X^6 - X^7 - X^8 \end{aligned} \quad (11-108)$$

因此，码字 WEF 由下式给出

$$A(X) = \frac{2X^3 + X^4 + X^5 + X^6 - X^7 - X^8}{1 - 2X - 2X^2 - X^3 + X^4 + X^5} = 2X^3 + 5X^4 + 15X^5 + \dots \quad (11-109)$$

因此该码包含两个重量为 3 的非零码字，五个重量为 4，十五个重量为 5 等。

采用本质上类似的过程可以得到关于码重量性质的其他信息。若这样添加修正状态图：相应于非零信息段的每个分支标以 W^ω ，其中 ω 是该分支上 k 个信息比特的重量，并用 L 标记每个分支，则码的码字输入输出重量枚举函数 (codeword input-output weight enumerating function, IOWEF) 由下式给出

$$A(W, X, L) = \sum_{w,d,l} A_{w,d,l} W^w X^d L^l \quad (11-110)$$

式中系数 $A_{w,d,l}$ 表示码重量为 d 、相应的信息权重是 w 、长度是 l 分支的码字数。图 11-1 中编码器的扩充后的修正状态图示于图 11-17 中。

例 11-12 (续)

对图 11-17 中的图，可以得到(见习题 11-18)

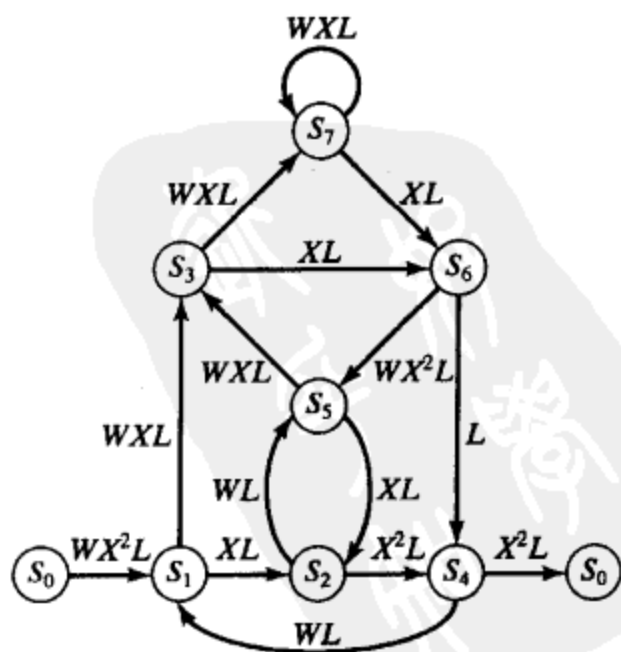


图 11-17 图 11-1 中 (2, 1, 3) 编码器扩充的修正状态图

$$\begin{aligned}
\Delta &= 1 - (X^8 W^4 L^7 + X^3 W^3 L^5 + X^7 W^3 L^6 + X^2 W^2 L^4 + X^4 W^4 L^7 + X^3 W^3 L^6 \\
&\quad + X^3 W L^3 + X W L^2 + X^5 W^3 L^4 + X^4 W^2 L^3 + X W L) + (X^4 W^4 L^7 \\
&\quad + X^8 W^4 L^7 + X^3 W^3 L^6 + X^3 W^3 L^5 + X^4 W^4 L^7 + X^8 W^4 L^7 + X^7 W^3 L^6 \\
&\quad + X^4 W^2 L^4 + X^2 W^2 L^3 + X^5 W^3 L^4) - (X^4 W^4 L^7 + X^8 W^4 L^7) \\
&= 1 - X W (L + L^2) - X^2 W^2 (L^4 - L^3) - X^3 W L^3 - X^4 W^2 (L^3 - L^4) \quad (11-111)
\end{aligned}$$

且

$$\begin{aligned}
\sum_i F_i \Delta_i &= X^{12} W^4 L^8 \cdot 1 + X^7 W^3 L^6 (1 - X W L^2) + X^{11} W^3 L^7 (1 - X W L) \\
&\quad + X^6 W^2 L^5 [1 - X W (L + L^2) + X^2 W^2 L^3] + X^8 W^4 L^8 \cdot 1 \\
&\quad + X^7 W^3 L^7 (1 - X W L) + X^7 W L^4 (1 - X W L - X^4 W^2 L^3) \\
&= X^6 W^2 L^5 + X^7 W L^4 - X^8 W^2 L^5 \quad (11-112)
\end{aligned}$$

因此, 码字 IOWEF 由下式给出

$$\begin{aligned}
A(W, X, L) &= \frac{X^6 W^2 L^5 + X^7 W L^4 - X^8 W^2 L^5}{1 - X W (L + L^2) - X^2 W^2 (L^4 - L^3) - X^3 W L^3 - X^4 W^2 (L^3 - L^4)} \\
&= X^6 W^2 L^5 + X^7 (X L^4 + W^3 L^6 + W^3 L^7) \\
&\quad + X^8 (W^2 L^6 + W^4 L^7 + W^4 L^8 + 2 W^4 L^9) + \dots \quad (11-113)
\end{aligned}$$

这意味着, 有一个重量为 6、长度为 5 个分支且信息重量为 2 的码字, 有三个重量为 7 的码字, 其中一个长度为 4 且信息重量为 1, 另一个长度为 6 个分支且信息重量为 3, 第三个长度为 7 个分支且信息重量为 3 等。 ■

$A(W, X, L)$ 中对应信息重量为 1 的项 $W L^{v+1}$, 总是代表穿过扩充的修正状态图的最短路径。然而有趣的是它可能不是最小重量路径。例 11-12 中, 例如项 $X^7 W L^4$ 代表最短路径, 而项 $X^6 W^2 L^5$ 代表最小重量路径。同样, 注意到以下这点是非常重要的: 码字 WEF $A(X)$ 是码的性质, 即它相对于编码器表示是不变的; 另一方面, 码字 IOWEF $A(W, X, L)$ 是编码器的性质, 因为它依赖于输入序列和码字之间的映射(见习题 11-21)。

IOWEF 的另一种形式, 仅包含关于输入和输出重量但不包含每个码字长度的信息, 可以通过简单地将 $A(W, X, L)$ 中的变量 L 设置为单位 1 而得到, 即

$$A(W, X) = \sum_{w,d} A_{w,d} W^w X^d = A(W, X, L) \Big|_{L=1} \quad (11-114)$$

式中 $A_{w,d} = \sum_i A_{w,d,i}$ 表示重量为 d 且信息重量为 w 的码字数。类似地, 对应于 IOWEF 的 WEF $A(X)$ 是

$$A(X) = \sum_d A_d X^d = A(W, X) \Big|_{W=1} = A(W, X, L) \Big|_{W=L=1} \quad (11-115)$$

其中 $A_d = \sum_w A_{w,d}$ 。在第 16 章中我们将会看到, 用于计算 turbo 码 WEF 的一个重要工具是码字条件重量枚举函数 (codeword conditional weight enumerating function, CWEF), 它计算和特定信息重量相联系的所有码字重量。对信息重量 ω , CWF 由下式给出

$$A_w(X) = \sum_d A_{w,d} X^d \quad (11-116)$$

根据公式(11-114)可以断定 IOWEF 可以表示成

$$A(W, X) = \sum_w W^w A_w(X) \quad (11-117)$$

显然, CWF 也是编码器的性质。 ■

例 11-12 (续)

公式(11-113)的简化 IOWEF 是

$$A(W, X) = A(W, X, L) \Big|_{L=1} = W^2 X^6 + (W + 2W^3) X^7 + (W^2 + 4W^4) X^8 + \dots, \quad (11-118)$$

表示有一个重量为6、信息重量为2的码字；有三个重量为7的码字，一个具有信息重量1，两个具有信息重量3，等等，我们可以得到 WEF

$$A(X) = A(W, X) \Big|_{W=1} = X^6 + 3X^7 + 5X^8 + \dots \quad (11-119)$$

它和前面在公式(11-106)中计算的 $A(X)$ 一致。

为计算该编码器的 CWF，我们必须找到和给定重量信息序列相联系的路径重量，它从初始状态到终止状态横贯修正状态图。存在这样一个重量为1的信息序列， $u = (1000)$ ；三个重量为2的序列 $u = (11000)$ ， (101000) 和 (1001000) ；九个重量为3的序列 $u = (111000)$ ， (1101000) ， (1011000) ， (10101000) ， (11001000) ， (10011000) ， (101001000) ， (100101000) 和 (1001001000) ，等等。（每个序列中最后三个0表示收尾比特。）因此，使用图 11-17，我们得到

$$A_1(X) = X^7 \quad (11-120a)$$

$$A_2(X) = X^6 + X^8 + X^{10} \quad (11-120b)$$

和

$$A_3(X) = 2X^7 + 3X^9 + 3X^{11} + X^{13} \quad (11-120c)$$

最后，使用公式(11-117)，我们可以写出

$$\begin{aligned} A(W, X) &= \sum_w W^w A_w(X) \\ &= WX^7 + W^2(X^6 + X^8 + X^{10}) + W^3(2X^7 + 3X^9 + 3X^{11} + X^{13}) + \dots \end{aligned} \quad (11-121)$$

由于采用不同的方法计算得到，因此不是显而易见公式(11-118)和(11-121)给出相同的结果，但是，如果我们在公式(11-118)中按照 W 升幂次序列出分子和分母来重写 $A(W, X, L)$ ，并令 $L=1$ ，我们得到

$$\begin{aligned} A(W, X) &= \frac{WX^7 + W^2(X^6 - X^8)}{1 - W(2X + X^3)} \\ &= WX^7 + W^2(X^6 + X^8 + X^{10}) + W^3(2X^7 + 3X^9 + 3X^{11} + X^{13}) + \dots \end{aligned}$$

和公式(11-121)的结果相同。

(11-122) ■

前面的例子说明了如果我们想要按照码字重量升幂的项来表示 IOWEF，我们必须在进行除法之前在 $A(W, X)$ 中按照 X 升幂的次序列出分子和分母项，如果我们想用信息重量升幂的项来表示 IOWEF，我们必须按照 W 的升幂列出它们。

现在考虑收尾编码器的情形，即编码器的输出序列限制为 $\lambda = h + m$ 个分组，其中 λ 代表输入序列的总长度，它由 h 个信息分组和 m 个收尾分组组成。为使前面计算 WEF 的过程和收尾编码器一致，我们按照 L 升幂的次序列出 IOWEF $A(W, X, L)$ 的分子和分母项。接着进行除法，并把结果幂级数中次数大于 L^A 的项舍去。

例 11-12 (续)

我们从把公式(11-113)重新写成如下形式开始

$$A(W, X, L) = \frac{L^4 WX^7 + L^5 W^2(X^6 - X^8)}{1 - LWX - L^2 WX - L^3[WX^3 + W^2(X^4 - X^2)] - L^4 W^2(X^2 - X^4)} \quad (11-123)$$

现在，令编码器在 $\lambda = 8$ 个分组之后终止，即考虑仅仅有 $h = 5$ 个信息分组和 $m = 3$ 个收尾分组的输入序列。接着，进行除法并舍去次数大于 L^8 的项得到(见习题 11-22)

$$\begin{aligned} A(W, X, L) &= L^4 WX^7 + L^5 W^2 X^6 + L^6(W^2 X^8 + W^3 X^7) + L^7[W^2 X^{10} + W^3(X^7 + X^{11}) + W^4 X^8] \\ &\quad + L^8[3W^3 X^9 + W^4(X^8 + X^{10} + X^{12}) + W^5 X^9] \end{aligned} \quad (11-124)$$

作为这个收尾编码器的 IOWEF。该 IOWEF 中刚好有 15 项，由于我们仅仅考虑第一个比特是 1 的信息序列，并且不考虑信息序列 $u = (10001000)$ ，因为它分叉于且再合并于状态 S_0 两次。（注意到 W^5 是公式(11-124)中的最大输入重量，因为 $\lambda - m = 5$ 。）我们现在可以分别计算收尾编码器的简化 IOWEF、CWEF、和 WEF 为

$$A(W, X) = A(W, X, L) \Big|_{L=1} = WX^7 + W^2(X^6 + X^8 + X^{10}) + W^3(2X^7 + 3X^9 + X^{11}) \\ + W^4(2X^8 + X^{10} + X^{12}) + W^5X^9 \quad (11-125a)$$

$$A_1(X) = X^7 \quad (11-125b)$$

$$A_2(X) = X^6 + X^8 + X^{10} \quad (11-125c)$$

$$A_3(X) = 2X^7 + 3X^9 + X^{11} \quad (11-125d)$$

$$A_4(X) = 2X^8 + X^{10} + X^{12} \quad (11-125e)$$

$$A_5(X) = X^9 \quad (11-125f)$$

以及

$$A(X) = A(W, X) \Big|_{W=1} = X^6 + 3X^7 + 3X^8 + 4X^9 + 2X^{10} + X^{11} + X^{12} \quad (11-125g)$$

注意到公式(11-125d)给出的收尾编码器的 CWEF $A_3(X)$ 不同于公式(11-120c)给出的不收尾编码器的表达式。这是由于在不收尾的情形下，一些重量为 $\omega = 3$ 的信息序列长度大于 $h = 5$ 比特，因此对于收尾编码器来说它们不是有效的输入序列。■

这里有必要注意到，前面用于计算码字 WEF 的方法仅考虑了在初始时间单元分叉于且以后再合并于全零状态刚好一次的非零码字。换言之，相同的延时码字或者分叉且再合并多于一次的码字没有计入。（这样计数非零码字重量是由于计算最大似然译码的卷积码的误比特率的原因，这个问题将在第 12 章中讨论。）因此，对于前面例子中考虑的信息序列长度 $h = 5$ 的收尾编码器的情形，仅仅算入了 15 个非零码字，而不是等价的 $(N, K^*) = [n(h + m), kh] = (16, 5)$ 分组码中存在的 $2^5 - 1 = 31$ 个非零码字。为计算 turbo 码的 WEF，需要计算通过收尾卷积编码器而得到的等价分组码的完整 WEF。这些方法将在第 16 章中更详细地讨论。

为计算由最大似然译码的卷积码的信息比特误码率——这是下一章将要讨论的内容，可以很方便地修改前面介绍的码字 WEF 表达式来表示和一个给定重量的码字相联系的非零信息比特数。这些修改的表达式称为比特 (bit) WEF。对信息重量 ω ，比特 WEF 由下式给出

$$B_\omega(X) = \sum_d B_{\omega,d} X^d \quad (11-126)$$

其中 $B_{\omega,d} = (\omega/k)A_{\omega,d}$ 。因此， $B_{\omega,d}$ 可以解释为和重量为 ω 的信息序列生成的重量为 d 的码字相联系的非零信息比特的总数被每个时间单元的信息比特数 k 相除。可以得出结论：比特 IOWEF 可以表示成

$$B(W, X) = \sum_{\omega,d} B_{\omega,d} W^\omega X^d = \sum_\omega W^\omega B_\omega(X) \quad (11-127)$$

比特 WEF 由下式给出

$$B(X) = \sum_d B_d X^d = B(W, X) \Big|_{W=1} \quad (11-128)$$

其中 $B_d = \sum_\omega B_{\omega,d}$ 等于和重量为 d 的码字相联系的非零信息比特的总数被每个时间单元的信息比特数 k 相除。这里，所有三个比特 WEF， $B_\omega(X)$ 、 $B(W, X)$ 和 $B(X)$ 都是编码器的性质。

我们现在注意到，这里给出的 $A_{\omega,d}$ 和 $B_{\omega,d}$ 的定义意味着，对任何 ω 和 d

$$B_d = \sum_\omega B_{\omega,d} = \sum_\omega (\omega/k) A_{\omega,d} \quad (11-129)$$

公式(11-129)可以用来确定以下的码字 IOWEF $A(W, X)$ 和比特 WEF $B(X)$ 之间的正式关系：

$$\begin{aligned}
 B(X) &= \sum_d B_d X^d = \sum_{w,d} (w/k) A_{w,d} X^d \\
 &= \frac{1}{k} \frac{\partial \left[\sum_{w,d} A_{w,d} W^w X^d \right]}{\partial W} \bigg|_{W=1} = \frac{1}{k} \frac{\partial A(W, X)}{\partial W} \bigg|_{W=1}
 \end{aligned} \quad (11-130)$$

即比特 WEF $B(X)$ 可以直接通过码字 IOWEF $A(W, X)$ 进行计算。

正如在下一章中将会看到的, 对卷积码在每个时间单元的 MLD 判决导致 k 个信息比特的译码。因此, 为计算每个信息比特的译码错误率, 我们在这里定义的比特 WEF 中包含了一个因子 $(1/k)$ 。对收尾卷积码, 在下一章中我们也将考虑最大后验概率译码。这种译码方法也用在第 16 章的 turbo 译码。其中每次对整个信息比特分组进行译码判决, WEF 的定义稍微有些不同。特别地, 修改了 IOWEF $A(W, X)$ 以包含等价 $(N, K^*) = [n(h+m), kh]$ 分组码中的所有 $2^{K^*} - 1 = 2^{kh} - 1$ 个非零码字, 包括延时后的给定码字和分叉于且再合并于所有非零状态不止一次的码字, 同时相关的码字 WEF 也相应进行了修改。同样, 如果 $K = k(h+m)$ 代表编码器输入比特的总数, 我们定义 $B_{w,d} = (\omega/K) A_{w,d}$, 对应的比特 WEF 也进行相应的修改。则 $B_{w,d}$ 代表和由重量为 ω 的信息序列产生的重量为 d 的码字相联系的非零信息比特的总数除以一个分组中输入比特总数 K 。(技术上, 由于 $K = k(h+m)$, 仅仅 K 个输入比特中的 kh 个是信息比特, 但是由于和 m 相比 h 通常很大, 我们一般忽略这种区别。)这些修正的 WEF 将在第 16 章中更详细地讨论。

例 11-12 (续)

在这个例子中, 把公式 (11-130) 运用到公式 (11-118) 给出的 $(2, 1, 3)$ 编码器的码字 IOWEF 上, 我们得到比特 WEF

$$\begin{aligned}
 B(X) &= \frac{1}{k} \frac{\partial A(W, X)}{\partial W} \bigg|_{W=1} = \frac{\partial [W^2 X^6 + (W + 2W^3) X^7 + (W^2 + 4W^4) X^8 + \dots]}{\partial W} \bigg|_{W=1} \\
 &= 2X^6 + 7X^7 + 18X^8 + \dots
 \end{aligned} \quad (11-131)$$

公式 (11-131) 中 $B(X)$ 的系数代表和该编码器每个重量的码字相联系的非零信息比特的总数。

我们将在下一章中看到, 怎样用比特 WEF $B(X)$ 来计算使用最大似然译码的卷积码的误比特率。

对系统编码器, 由于 k 个信息序列和 $(n-k)$ 个奇偶序列构成一个码字, 码字的重量是信息重量 ω 加上奇偶序列的重量。这里, 可以很方便地把码字 CWF 表示成

$$A_w(Z) = \sum_z A_{w,z} Z^z \quad (11-132)$$

其中, $A_{w,z}$ 是信息重量为 ω 且校验重量 (parity weight) 为 z 的码字数。(这里, 我们注意到信息重量为 ω 且校验重量为 z 的码字数 $A_{w,z}$ 和信息重量为 ω 且总重 $d = z + \omega$ 的码字数 $A_{w,d}$ 相同。)则码字输入冗余重量枚举函数 (codeword input redundancy weight enumerating function, IRWEF) 由下式给出

$$A(W, Z, L) = \sum_{w,z,l} A_{w,z,l} W^w Z^z L^l \quad (11-133)$$

式中 $A_{w,z,l}$ 是长为 l 的信息重量为 ω 且校验重量为 z 的码字数, 则

$$A(W, Z) = \sum_{w,z} A_{w,z} W^w Z^z = A(W, Z, L) \bigg|_{L=1} = \sum_{w,z} W^w A_w(Z) \quad (11-134)$$

可以断定码字 WEF 可以表示成

$$A(X) = \sum_d A_d X^d = A(W, Z) \bigg|_{W=Z=X} \quad (11-135)$$

式中 $A_d = \sum_{\omega+z=d} A_{\omega,z}$ 是重量为 d 的码字总数。(符号 $\sum_{\omega+z=d} A_{\omega,z}$ 意味着求和遍历所有满足 $\omega + z = d$ 的系数 $A_{\omega,z}$ 。)系统编码器的比特 WEF 用类似公式(11-126) ~ (11-128)的方法进行计算, 其中 $B_{\omega,z} = (\omega/k)A_{\omega,z}$ 。(或者对收尾编码器 $B_{\omega,z} = (\omega/K)A_{\omega,z}$)。对信息重量 ω , 比特 CWF 由下式给出

$$B_w(Z) = \sum_z B_{w,z} Z^z \quad (11-136)$$

式中 $B_{w,z}$ 可以解释为, 和校验重量为 z 且信息重量为 w 的码字相联系的非零信息比特的总数除以每个时间单元的信息比特数 k 。可以直接断定比特 IRWEF 能表示成

$$B(W, Z) = \sum_{w,z} B_{w,z} W^w Z^z = \sum_w W^w B_w(Z) \quad (11-137)$$

以及比特 WEF 由下式给出

$$B(X) = \sum_d B_d X^d = B(W, Z) \Big|_{W=Z=X} \quad (11-138)$$

式中 $B_d = \sum_{\omega+z=d} B_{\omega,z}$ 是和重量为 d 的码字相联系的非零信息比特总数除以每个时间单元的信息比特数 k 。最后, 利用公式(11-137)和(11-138)可以建立下面的码字 IRWEF $A(W, Z)$ 和比特 WEF $B(X)$ 之间的正式关系:

$$\begin{aligned} B(X) &= B(W, Z) \Big|_{W=Z=X} = \sum_{w,z} B_{w,z} W^w Z^z \Big|_{W=Z=X} = \sum_{w,z} (\omega/k) A_{\omega,z} W^w Z^z \Big|_{W=Z=X} \\ &= \frac{1}{k} W \frac{\partial [\sum_{w,z} A_{\omega,z} W^w Z^z]}{\partial W} \Big|_{W=Z=X} = \frac{1}{k} W \frac{\partial A(W, Z)}{\partial W} \Big|_{W=Z=X} \end{aligned} \quad (11-139)$$

即比特 WEF $B(X)$ 可以直接通过码字 IRWEF $A(W, Z)$ 计算。

再一次地, 对系统编码器, 码字 WEF 是码的一个性质, 而码字 CWF 和 IRWEF、比特 CWF 和 IRWEF 以及 WEF 都是编码器的性质。

为使用这里介绍的传递函数方法来计算系统编码器的 IRWEF, 在扩充的修正状态图中, 我们使用代表校验重量的标记 Z^z 代替代表码字重量的标记 X^d 。我们现在用一个系统反馈编码器的例子来说明对系统编码器计算 WEF 的方法。首先, 我们注意到对反馈编码器, 用于对编码器进行收尾的比特通常是非零的。因此, 这种情形下, 我们认为 WEF 中 W 的幂次是非零输入比特而不是非零信息比特, 因为收尾比特不被考虑成信息比特, 但是这种差别仅仅对低重量输入序列来说是重要的。 ■

例 11-14 计算(2, 1, 2)系统反馈编码器的 WEF

考虑一个(2, 1, 2)系统反馈卷积编码器, 其生成矩阵由下式给出

$$G(D) = [1 \quad (1 + D^2)/(1 + D + D^2)] \quad (11-140)$$

该编码器的控制器规范形式实现和扩充的修正状态图示于图 11-18 中。仿照类似于例 11-12 和 11-13 的过程, 我们可以看到在图 11-18b 的图中有三个环路,

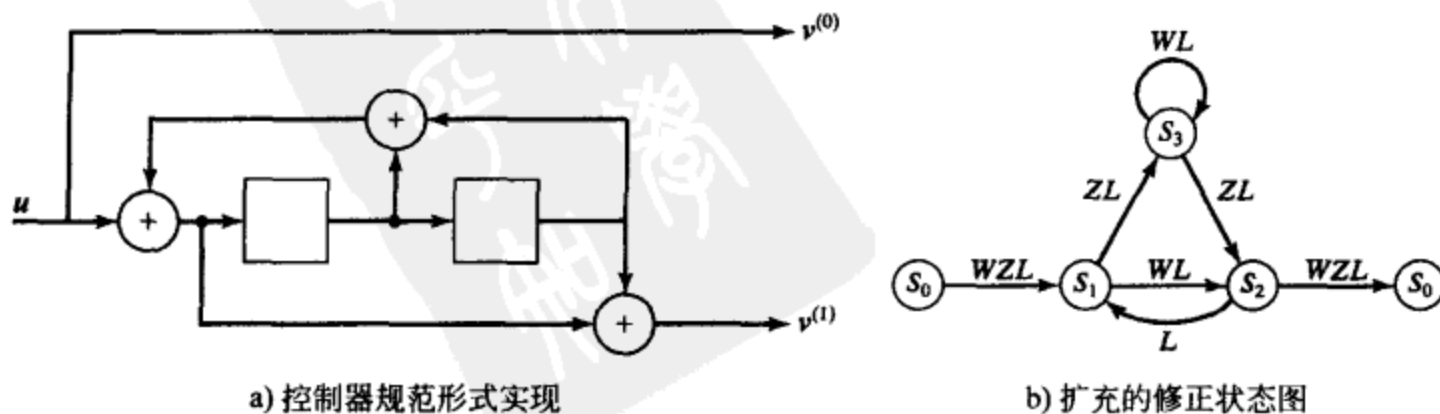


图 11-18 (2, 1, 2) 系统反馈编码器

环路 1: $S_1 S_3 S_2 S_1$ ($C_1 = Z^2 L^3$)

环路 2: $S_1 S_2 S_1$ ($C_2 = WL^2$)

环路 3: $S_3 S_3$ ($C_3 = WL$)

我们也可以看到有一对不相连环路:

环路对 1: (环路 2, 环路 3) ($C_2 C_3 = W^2 L^3$)

这里没有其他不相连环路的集合。因此,

$$\Delta = 1 - (Z^2 L^3 + WL^2 + WL) + (W^2 L^3) = 1 - Z^2 L^3 - W(L + L^2) + W^2 L^3 \quad (11-141)$$

现在我们注意到在图 11-18b 的图中有两条前向路径:

前向路径 1: $S_0 S_1 S_2 S_0$ ($F_1 = W^3 Z^2 L^3$)

前向路径 2: $S_0 S_1 S_3 S_2 S_0$ ($F_2 = W^2 Z^4 L^4$)

不接触前向路径 1 的子图简单地就是围绕状态 S_3 的单分支环路, 即环路 3。因此

$$\Delta_1 = 1 - WL \quad (11-142)$$

前向路径 2 接触图中的所有状态, 因此

$$\Delta_2 = 1 \quad (11-143)$$

最后, 运用公式(11-99), 我们得到码字 IRWEF:

$$\begin{aligned} A(W, Z, L) &= \frac{W^3 Z^2 L^3 (1 - WL) + W^2 Z^4 L^4 \cdot 1}{1 - Z^2 L^3 - W(L + L^2) + W^2 L^3} = \frac{W^2 Z^4 L^4 + W^3 Z^2 L^3 - W^4 Z^2 L^4}{1 - Z^2 L^3 - W(L + L^2) + W^2 L^3} \\ &= \frac{W^2 Z^4 L^4}{F(Z, L)} + W^3 \left[\frac{Z^2 L^3}{F(Z, L)} + \frac{Z^4 (L^5 + L^6)}{F^2(Z, L)} \right] + \dots \end{aligned} \quad (11-144)$$

其中定义 $F(Z, L) = 1 - Z^2 L^3$ 。这里我们看到, 存在无限数量的码字对应于重量为 2 的输入序列。(状态图中, 环路 1 有零输入重量与之相联系, 因此无限数量的输入重量为 2 的码字可以通过在任意时间数中遍历该环路而生成。)我们可以通过扩充公式(11-144)中的第一项来表示这些码字如下:

$$\frac{W^2 Z^4 L^4}{F(Z, L)} = W^2 (Z^4 L^4 + Z^6 L^7 + Z^8 L^{10} + \dots) \quad (11-145)$$

即, 对应于重量为 2 的输入序列, 由于环路 1 的增益 $C_1 = Z^2 L^3$, 存在具有校验重量 $4 + 2j$ 和长度 $4 + 3j$ 的码字, $j = 0, 1, 2, \dots$ 。类似地, 存在无限数量的码字对应于重量为 3 的输入序列等。

现在我们从公式(11-144)得到不包含长度信息的 IRWEF 如下:

$$A(W, Z) = A(W, Z, L) \Big|_{L=1} = \frac{W^2 Z^4 + W^3 Z^2 - W^4 Z^2}{1 - Z^2 - 2W + W^2} = \frac{W^2 Z^4}{F(Z)} + W^3 \left[\frac{Z^2}{F(Z)} + \frac{2Z^4}{f^2(Z)} \right] + \dots \quad (11-146)$$

式中 $F(Z) = 1 - Z^2$, 我们还可以利用公式(11-134)得到如下的 CWF:

$$A_2(Z) = \frac{Z^4}{1 - Z^2} \quad (11-147a)$$

$$A_3(Z) = \frac{Z^2}{1 - Z^2} + \frac{2Z^4}{1 - 2Z^2 + Z^4} \quad (11-147b)$$

等等。有趣地注意到, 这里公式(11-147a)意味着所有输入重量为 2 的码字至少有总重量 6 (输入重量 2 加上至少校验重量 4), 而公式(11-147b)意味着存在一个输入重量为 3 的码字, 其总重量仅为 5。因此, 这里最小重量码字由一个重量为 3 的输入序列产生。最后, 码字 WEF 通过在公式(11-146)中令 $W = Z = X$ 而得到, 给出其表达式如下

$$\begin{aligned}
 A(X) &= A(W, Z) \Big|_{W=Z=X} = \frac{X^6 + X^5 - X^6}{1 - X^2 - 2X + X^2} \\
 &= \frac{X^5}{1 - 2X} = X^5 + 2X^6 + 4X^7 + 8X^8 + \dots
 \end{aligned} \quad (11-148)$$

(注意到为了增加输入重量, 置换 $W = Z = X$ 直接代入到公式(11-146)的有理函数表示式而不是其级数表示中, 因为这导致了 $A(X)$ 的简化表达式。)我们从公式(11-148)看出, 该码包含一个重量为 5 的码字, 两个重量为 6, 四个重量为 7, 等等。习题 11-23 显示了和公式(11-140)的系统反馈编码器产生相同码的等价非系统前馈编码器, 具有和公式(11-148)相同的码字 WEF $A(X)$ 。但是其 IOWEF 和 CWF 代表和前面计算的 IRWF 和 CWF 不同的输入输出关系, 因为这些函数是编码器的性质。

最后, 仿照公式(11-136) ~ (11-138), 比特 CWF 由下式给出

$$B_2(Z) = \frac{2Z^4}{1 - Z^2} \quad (11-149a)$$

$$B_3(Z) = \frac{3Z^2}{1 - Z^2} + \frac{6Z^4}{1 - 2Z^2 + Z^4} \quad (11-149b)$$

比特 IRWF 由下式给出

$$B(W, Z) = \sum_w W^w B_w(Z) = \frac{2W^2 Z^4}{1 - Z^2} + \frac{3W^3 Z^2}{1 - Z^2} + \frac{6W^3 Z^4}{1 - 2Z^2 + Z^4} + \dots \quad (11-150)$$

比特 WEF 由下式给出

$$B(X) = B(W, Z) \Big|_{W=Z=X} = \frac{3X^5}{1 - X^2} + \frac{2X^6}{1 - X^2} + \frac{6X^7}{1 - 2X^2 + X^4} + \dots \quad (11-151)$$

我们也可以使用公式(11-139)计算 $B(X)$ 如下:

$$\begin{aligned}
 B(X) &= \frac{1}{k} W \frac{\partial A(W, Z)}{\partial W} \Big|_{W=Z=X} = W \frac{\partial [(W^2 Z^4 + W^3 Z^2 - W^4 Z^2)/(1 - Z^2 - 2W + W^2)]}{\partial W} \Big|_{W=Z=X} \\
 &= \frac{N(W, Z)}{D(W, Z)} \Big|_{W=Z=X}
 \end{aligned} \quad (11-152)$$

其中

$$N(W, Z) = 2W^2(Z^4 - Z^6) + W^3(3Z^2 - 5Z^4) - 2W^4(Z^2 - 2Z^4 + 3Z^2) + 7W^5 Z^2 - 2W^6 Z^2 \quad (11-153a)$$

以及

$$D(W, Z) = 1 - 2Z^2 + Z^4 - 4W(1 - Z^2) + 2W^2(3 - Z^2) - 4W^3 + W^4 \quad (11-153b)$$

现在, 进行除法运算, 我们得到(见习题 11-24)

$$B(X) = 3X^5 + 6X^6 + 14X^7 + 32X^8 + \dots \quad (11-154)$$

即, 重量为 5 的码字有信息重量 3, 两个重量为 6 的码字有总信息重量 6, 等等。(因为这里 $k=1$, $B(X)$ 的系数代表和一个给定重量的所有码字相关的总信息重量。)注意到似乎公式(11-151)和(11-154)不会给出相同的结果, 因为公式(11-151)中显示的三个项总体上是基于输入重量 2 和 3 的。如果也包含输入重量 4、5 和 6, 则公式(11-151)和(11-154)中 $B(X)$ 的系数一直到 X^8 都是一致的。通常, 基于公式(11-139)的方法给出最简单的计算 $B(X)$ 的方式。■

在例 11-14 中, 我们看到在状态图中有一个零输入重量环路, 它允许编码器以重量 2 的输入产生无限数目的码字。这些零输入重量环路是所有反馈编码器的典型特征, 其中单个 1 输入驱使编码器离开全零状态, 接着在零输入重量环路被遍历某个次数之后, 另一个 1 输入使编码器回到全零状态。在第 16 章中我们将看到, 由这些重量 2 输入序列产生的码字在 turbo 码的性能中起了决定性作用。

我们现在引入另一种计算卷积编码器的 WEF 的方法。前述的基于把编码器的状态图表示成信号流图的方法, 具有方便地对分叉且再合并于全零状态恰好一次的所有码字的集合可视化的优点。但是在计算上, 一旦整体约束长度 v 超过 3, 即状态图包含多于 8 个状态, 它便难以使用。一种更有效的计算编码器传递函数的方式是使用编码方程的状态变量表达形式。我们通过继续前述的例子来说明这种方法。

例 11-14 (续)

在图 11-18 的扩充的修正状态图中, 令 Σ_i 是代表从初始状态 S_0 到状态 S_i 的所有路径重量的状态变量, $i=1, 2, 3$ 。接着, 我们可以写出下列状态方程组:

$$\Sigma_1 = WZL + L\Sigma_2 \quad (11-155a)$$

$$\Sigma_2 = WL\Sigma_1 + ZL\Sigma_3 \quad (11-155b)$$

$$\Sigma_3 = ZL\Sigma_1 + WL\Sigma_3 \quad (11-155c)$$

图的整体传递函数由下式给出

$$A(W, Z, L) = WZL\Sigma_2 \quad (11-156)$$

公式(11-155)是一组以状态变量 Σ_1 , Σ_2 和 Σ_3 表示的线性方程组。我们可以把它们写成矩阵形式

$$\begin{bmatrix} WZL \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & -L & 0 \\ -WL & 1 & -ZL \\ -ZL & 0 & 1-WL \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \\ \Sigma_3 \end{bmatrix} \quad (11-157)$$

对状态变量 Σ_2 , 使用克拉默(Cramer)法则来解方程(11-157), 我们得到

$$\Sigma_2 = \frac{\det \begin{bmatrix} 1 & WZL & 0 \\ -WL & 0 & -ZL \\ -ZL & 0 & 1-WL \end{bmatrix}}{\det \begin{bmatrix} 1 & -L & 0 \\ -WL & 1 & -ZL \\ -ZL & 0 & 1-WL \end{bmatrix}} = \frac{WZ^3L^3 + W^2ZL^2 - W^3ZL^3}{1 - Z^2L^3 - W(L + L^2) + W^2L^3} \quad (11-158)$$

现在, 我们运用公式(11-156)得到 IRWEF:

$$A(W, Z, L) = WZL\Sigma_2 = \frac{W^2Z^4L^4 + W^3Z^2L^3 - W^4Z^2L^4}{1 - Z^2L^3 - W(L + L^2) + W^2L^3} \quad (11-159)$$

它和前面使用信号流图方法得到的结果相同。

从前述的例子中可以很明显看出, 当整体约束长度 v 变得很大时, 使用传递函数方法计算卷积码的 WEF 很快变得不现实了。但是正如在下一节中将会看到的, 为估计在中等到高 SNR 时码性能所必需的重要的距离特性可以不通过传递函数而进行计算。但是, 传递函数仍然是一个重要的概念性工具, 在第 12、16 和 18 章中将会使用它来求得采用最大似然译码的卷积码的性能界。

11.3 卷积码的距离特性

卷积码的性能取决于所采用的译码算法和码的距离特性。本节将引入几种卷积码的距离度量, 它们和码性能的关系将在后面的章节中讨论。

卷积码的最重要的距离度量是最小自由距离 (free distance) d_{free} 。

定义 11-7 卷积码的最小自由距离定义为

$$d_{\text{free}} \triangleq \min_{u', u''} \{d(v', v'') : u' \neq u''\} \quad (11-160)$$

其中 v' 和 v'' 分别是信息序列 u' 和 u'' 对应的码字。

在公式(11-160)中, 假定码字 v' 和 v'' 具有有限长度, 起始并且终止于全零状态 S_0 , 即在信息序列 u' 和 u'' 上附加 m 个收尾分组。如果 u' 和 u'' 具有不同的长度, 则在较短的序列中添加一些零以保证对应的码字具有相同的长度。因此, d_{free} 是码中任意两个有限长度码字之间的最小距离。由于卷积码是线性码,

$$d_{\text{free}} = \min_{u', u''} \{w(v' + v'') : u' \neq u''\} = \min_u \{w(v) : u \neq 0\} = \min_u \{w(uG) : u \neq 0\} \quad (11-161)$$

其中 v 是对应于信息序列 u 的码字。因此, d_{free} 是由任意有限长度的非零信息序列产生的最小重量码字。它也是状态图中分叉且再合并于全零状态 S_0 的所有有限长度路径的最小重量, 因此它是 WEF $A(X)$ 中 X 的最低次幂。例如, 对例 11-12 中的 $(2, 1, 3)$ 码有 $d_{\text{free}} = 6$, 对例 11-13 中的 $(3, 2, 3)$ 码有 $d_{\text{free}} = 3$ 。

卷积码的另一个重要的距离度量是列距离函数 (column distance function, CDF)。令

$$[v]_l = (v_0^{(0)} v_0^{(1)} \cdots v_0^{(n-1)}, v_1^{(0)} v_1^{(1)} \cdots v_1^{(n-1)}, \cdots, v_l^{(0)} v_l^{(1)} \cdots v_l^{(n-1)}) \quad (11-162)$$

表示码字 v 的第 l 级截短, 令

$$[u]_l = (u_0^{(1)} u_0^{(2)} \cdots u_0^{(k)}, u_1^{(1)} u_1^{(2)} \cdots u_1^{(k)}, \cdots, u_l^{(1)} u_l^{(2)} \cdots u_l^{(k)}) \quad (11-163)$$

表示信息序列 u 的第 l 级截短。

定义 11-8 l 阶列距离函数 d_l 定义为

$$d_l \triangleq \min_{[u']_l, [u'']_l} \{d([v']_l, [v'']_l) : [u']_0 \neq [u'']_0\} = \min_{[u]_l} \{w[v]_l : [u]_0 \neq 0\} \quad (11-164)$$

其中 v 是对应于信息序列 u 的码字。

因此, d_l 是在最初 $(l+1)$ 个时间单元上的最小重量码字, 码字的最初信息分组是非零。根据码的生成矩阵

$$[v]_l = [u]_l [G]_l \quad (11-165)$$

式中 $[G]_l$ 是由 G (见公式(11-22)) 的最初 $n(l+1)$ 列构成的一个 $k(l+1) \times n(l+1)$ 矩阵, 它具有如下形式:

$$[G]_l = \begin{bmatrix} G_0 & G_1 & \cdots & G_l \\ & G_0 & \cdots & G_{l-1} \\ & & \ddots & \vdots \\ & & & G_0 \end{bmatrix}, \quad l \leq m \quad (11-166a)$$

$$[G]_l = \begin{bmatrix} G_0 & G_1 & \cdots & G_{m-1} & G_m \\ G_0 & \cdots & G_{m-2} & G_{m-1} & G_m \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & G_1 & G_2 & G_3 & \cdots & G_m \\ & G_0 & G_1 & G_2 & \cdots & G_{m-1} & G_m \\ & & G_0 & G_1 & \cdots & G_{m-2} & G_{m-1} \\ & & & G_0 & \vdots & \vdots & \vdots \\ & & & & \ddots & G_1 & G_2 \\ & & & & & G_0 & G_1 \\ & & & & & & G_0 \end{bmatrix}, \quad l > m \quad (11-166b)$$

则

$$d_l = \min_{[u]_l} \{w([u]_l [G]_l) : [u]_0 \neq 0\} \quad (11-167)$$

看起来仅仅依赖于 G 的前 $n(l+1)$ 列。这说明了其名字“列距离函数”的由来。

这里我们注意到, 公式(11-166)中使用的生成矩阵 G 形式(也在公式(11-22)中)假定为前馈编码器实现, 即存储级数为 m 的多项式生成矩阵。对具有有理函数生成矩阵的反馈编码器, 相应的生成序列具有无限持续时间, $k \times n$ 维子矩阵 G_l 无限扩展为公式(11-166)的右端。这里仍可以使用公式(11-167)来表示 CDF。我们也注意到自由距离 d_{free} 是独立于编码器实现的, 即它是码的性质, 而列距离函数 d_l 是编码器的性质; 但是, 如果我们把注意力集中在 $k \times n$ 维子矩阵 G_0 满秩的编码器上, 则 CDF d_l 是独立于编码器实现的(这导致了一种合理的性质: 相对于信息序列, 码字没有被延迟)。因此, 我们把 CDF d_l 看成是码的性质。

定义 11-8 意味着, d_l 不会随着 l 的增加而降低, 即它是 l 的单调非递减函数。计算码的 CDF 的最有效的方法是修正一种在第 13 章引入的序列译码算法(见第 13.4 节)。 $G(D) = [1 + D + D^2 + D^5 + D^6 + D^8 + D^{13} + D^{16}, 1 + D^3 + D^4 + D^7 + D^9 + D^{10} + D^{11} + D^{12} + D^{14} + D^{15} + D^{16}]$ 的 $(2, 1, 16)$ 非系统码的完整 CDF 示于图 11-9 中。

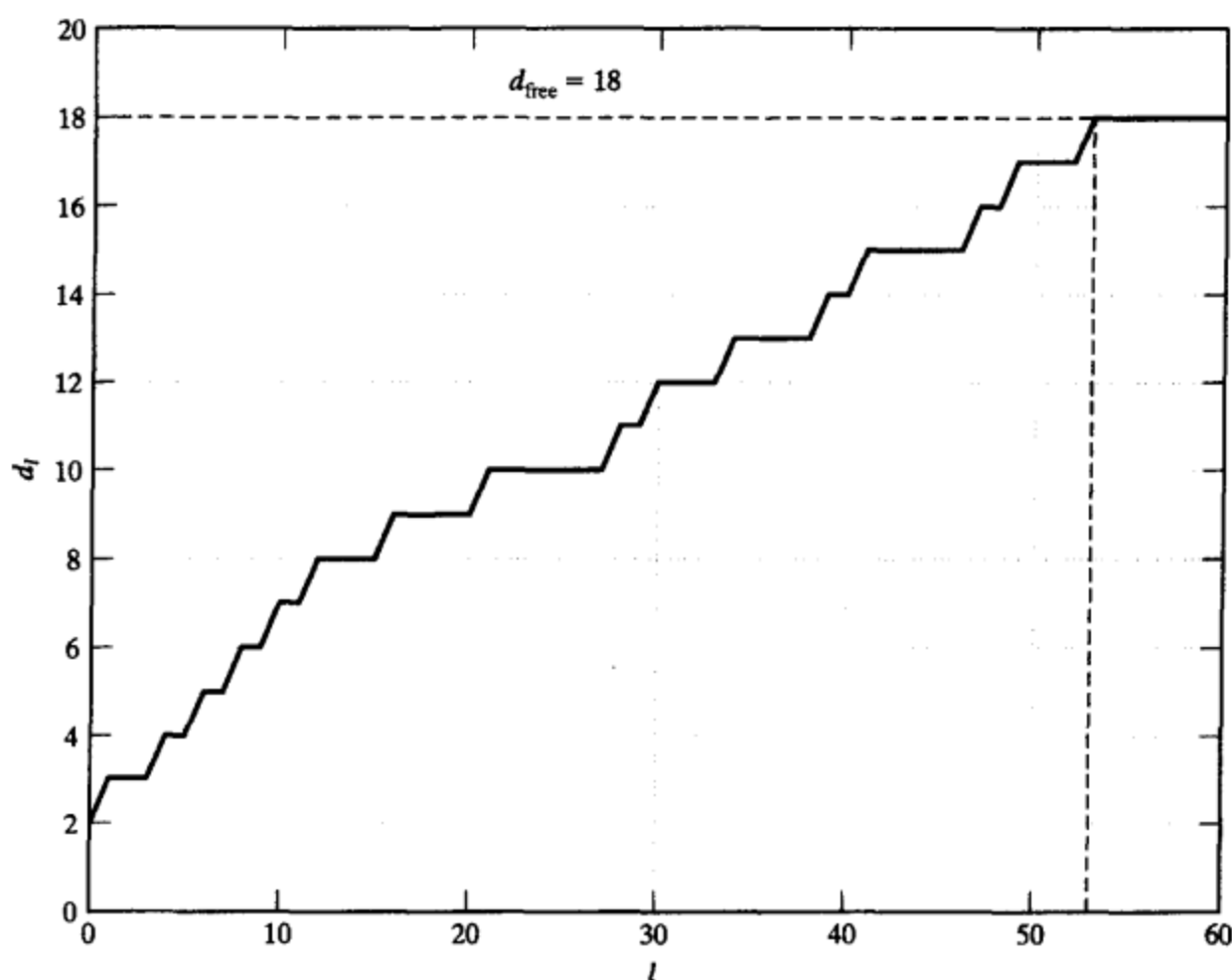


图 11-19 $(2, 1, 16)$ 码的列距离函数

有两种情形具有特殊的意义: $l = m$ 和 $l \rightarrow \infty$ 。对 $l = m$, d_m 称为卷积码的最小距离 (minimum distance)。这里 d_m 也表示成 $d_{\min}$ 。从公式(11-167)我们看出, $d_{\min}$ 代表在最初 $(m+1)$ 个时间单元上的最小重量码字, 码字的最初信息分组是非零。对图 11-19 中的码有 $d_{\min} = d_{16} = 9$ 。许多关于卷积码的早期著作把 $d_{\min}$ 看成是最重要的距离参数, 因为那时的主要译码技术具有 $(m+1)$ 时间单元的译码存储。(见第 13.5 节卷积码的大数逻辑译码。)最近, 由于最大似然 (ML) 译码、最大后验 (MAP) 译码和序列译码变得更为主要, d_{free} 和 CDF 已经取代 $d_{\min}$ 成为主要的距离参数, 因为这些技术的译码存储是无限的。在第 12 章和第 13 章中有关于距离度量和译码算法间关系的详尽讨论。

对 $l \rightarrow \infty$, $\lim_{l \rightarrow \infty} d_l$ 是任意长度的初始信息分组为非零的最小重量码字。比较 $\lim_{l \rightarrow \infty} d_l$ 和 d_{free} 的定义, 我们可以得出(见习题 11-31)对非恶性编码器有

$$\lim_{l \rightarrow \infty} d_l = d_{\text{free}} \quad (11-168)$$

这里, d_l 逐渐达到 d_{free} , 然后保持恒定。这通常在 l 大约为 $3v$ 时发生。对图 11-19 中的码, 当 $l \geq 53$ 时有 $v = 16$, $d_l = 18$, 因此 $d_{\text{free}} = 18$ 。然而由于 d_{free} 是针对有限长度码字定义的, 对于恶性编码器公式(11-168)不一定成立。■

例 11-10 (续)

再一次考虑其状态图示于图 11-14 中的恶性编码器。对该编码器有 $d_0 = 2$, 以及 $d_1 = d_2 = \dots = \lim_{l \rightarrow \infty} d_l = 3$, 因为即使在 $l \rightarrow \infty$ 的极限时, 截短信息序列 $[u]_l = (1, 1, 1, \dots, 1)$ 也总是产生截短码字 $[v]_l = (11, 01, 00, 00, \dots, 00)$ 。但是注意到, 状态图中分叉且再合并于全零状态 S_0 的所有有限长度路径至少具有重量 4, 因此 $d_{\text{free}} = 4$ 。这种情况下我们有 $\lim_{l \rightarrow \infty} d_l = 3 \neq d_{\text{free}} = 4$, 即不满足公式(11-168)。■

恶性编码器的典型特征是无限重量信息序列产生有限重量码字。在某些情形下, 如前面的例子中, 由于状态图中的零输出重量环路, 码字的重量可能小于码的自由距离。换言之, 一个永远围绕零输出重量环路绕行的信息序列, 即使耗尽其无限重量也不会增加码字的重量。在非恶性编码器情形下, 除了环绕状态 S_0 的零重量环路之外没有其他的零输出重量环路, 故所有无限重量的信息序列必定产生无限重量的码字, 且最小重量码字总是具有有限长度。不幸的是, 有些情况下产生最小重量码字的信息序列可能相当长, 因而计算 d_{free} 是一个艰巨的任务。

在给定码率 R 和整体约束长度 v 时, 一个卷积码可实现的最优 d_{free} 通常还不能精确得出, 但是 d_{free} 的上下限可以通过随机编码方法而得出。参考文献[16]、[17]和[18]中详尽地讨论了这些限。对非系统编码器和系统编码器的限进行比较表明, 在给定码率和约束长度下, 非系统前馈编码器可以比系统前馈编码器有更大的自由距离。这个结论将被下面两章介绍的码的构造性结果所证实。它的一个重要推论是, 具有大的 d_{free} 的码一定会被选用于 ML、MAP 或序列译码。因此, 当需要一个系统编码器实现时, 通常较好的方法是选择 d_{free} 较大的非系统前馈编码器并将其转化成等价的系统反馈编码器。

习题

11-1 考察 $g^{(0)} = (1\ 1\ 0)$, $g^{(1)} = (1\ 0\ 1)$, $g^{(2)} = (1\ 1\ 1)$ 的 $(3, 1, 2)$ 非系统前馈编码器,

1) 画出编码器结构框图。

2) 计算时域生成矩阵 G 。

3) 计算信息序列 $u = (1\ 1\ 1\ 0\ 1)$ 对应的码字 v 。

11-2 考察图 11-3 所示的 $(4, 3, 3)$ 非系统前馈编码器。

1) 计算该编码器的生成序列。

2) 计算时域生成矩阵 G 。

3) 计算信息序列 $u = (110, 011, 101)$ 对应的码字 v 。

11-3 考察习题 11-1 中的 $(3, 1, 2)$ 编码器。

1) 计算变换域生成矩阵 $G(D)$ 。

2) 计算输出序列集合 $V(D)$, 以及信息序列 $u(D) = 1 + D^2 + D^3 + D^4$ 对应的码字 $v(D)$ 。

11-4 考察图 11-2 所示的 $(3, 2, 2)$ 非系统前馈编码器。

1) 计算复合生成多项式 $g_1(D)$ 和 $g_2(D)$ 。

2) 计算输入信息序列集合 $U(D) = [1 + D + D^3\ 1 + D^2 + D^3]$ 对应的码字 $v(D)$ 。

11-5 考察

$$g^{(1)} = (1\ 0\ 1\ 1\ 0\ 1),$$

$$g^{(2)} = (1\ 1\ 0\ 0\ 1\ 1).$$

的(3, 1, 5)系统前馈编码器,

1) 计算时域生成矩阵 G 。

2) 计算信息序列 $u = (1\ 1\ 0\ 1)$ 对应的奇偶序列 $v^{(1)}$ 和 $v^{(2)}$ 。

11-6 考察

$$g_1^{(2)}(D) = 1 + D^2 + D^3,$$

$$g_2^{(2)}(D) = 1 + D + D^3.$$

的(3, 2, 3)系统前馈编码器,

1) 画出该编码器的控制器规范形式实现。在该实现中需要多少延时单元?

2) 画出仅需要三个延时单元的简化观测器规范形式实现。

11-7 验证把公式(11-34)和(11-70)的非系统前馈实现变换到公式(11-66)和(11-71)的系统反馈实现的初等行变换序列。

11-8 画出公式(11-64)中生成矩阵 $G'(D)$ 的观测器规范形式实现并确定其整体约束长度 v 。

11-9 考虑一个码率为 $R = 2/3$ 的非系统前馈编码器, 其生成矩阵由下式给出

$$G(D) = \begin{bmatrix} D & D & 1 \\ 1 & D^2 & 1 + D + D^2 \end{bmatrix}.$$

1) 画出 $G(D)$ 的控制器规范形式实现。整体约束长度 v 是多少?

2) 找出等价的系统反馈编码器的生成矩阵 $G'(D)$ 。 $G'(D)$ 可实现吗? 如果不是, 找出一个等价的可实现生成矩阵并画出相应的最小编码器实现。这个最小实现是控制器规范形式还是观测器规范形式? 最小整体约束长度 v 是多少?

11-10 使用初等行变换把公式(11-77)的码率 $R = 2/3$ 生成矩阵转化成系统反馈形式, 并画出最小观测器规范形式编码器实现。找到并画出一个具有相同状态数的非系统反馈控制器规范形式编码器实现。

11-11 使用公式(11-82)中的符号和图 11-11 中的标记方法重画图 11-7b 中(3, 2, 2)系统反馈编码器的观测器规范形式实现。

11-12 考虑图 11-6c 所示的(3, 1, 2)系统反馈编码器。当信息序列 $u = (1\ 0\ 1\ 1\ 1)$ 时, 确定使该编码器回到全零状态所必需的 $v = 2$ 个收尾比特。

11-13 考虑图 11-3 中控制器规范形式的(4, 3, 3)非系统前馈编码器实现。

1) 画出等价的观测器规范形式的非系统前馈编码器实现, 并确定使该编码器回到全零状态所必需的收尾比特数。该编码器实现的整体约束长度是多少?

2) 现在, 确定等价的观测器规范形式的系统反馈编码器实现, 并计算使该编码器回到全零状态所必需的收尾比特数。该编码器实现的整体约束长度是多少?

11-14 考虑 $G(D) = [1 + D^2\ 1 + D + D^2]$ 的(2, 1, 2)非系统前馈编码器

1) 计算该生成多项式的 GCD。

2) 计算它的最小延时前馈逆的传递函数矩阵 $G^{-1}(D)$ 。

11-15 考虑 $G(D) = [1 + D^2\ 1 + D + D^2 + D^3]$ 的(2, 1, 3)非系统前馈编码器

1) 计算该生成多项式的 GCD。

2) 画出编码器状态图。

3) 在状态图中找到一个零输出重量环路。

4) 计算产生有限重量码字的无限重量信息序列。

5) 这个编码器是恶性的还是非恶性的?

11-16 对 (n, k, v) 系统编码器的前馈逆, 计算其传递函数矩阵 $G^{-1}(D)$ 的一般形式。最小延时 l 是多少?

11-17 验证例 11-13 中 WEF 的计算。

11-18 验证例 11-12 中 IOWEF 的计算。

11-19 考虑习题 11-1 中的(3, 1, 2)编码器

1) 画出编码器的状态图。

2) 画出编码器的修正状态图。

3) 计算 WEF $A(X)$ 。

- 4) 画出编码器的扩充的修正状态图。
 - 5) 计算 $IOWEF A(W, X, L)$ 。
- 11-20 使用合适的软件, 对图 11-3 中的 (4, 3, 3) 编码器重复习题 11-18。
- 11-21 对例 11-1, 考虑通过每个生成多项式被 $g^{(0)}(D) = 1 + D^2 + D^3$ 除得到的等价系统反馈编码器。
- 1) 画出该编码器的扩充的修正状态图。
 - 2) 计算该编码器的 $IRWEF A(W, Z)$, 两个最低输入重量 $CWEF$ 和 $WEF A(X)$ 。
 - 3) 将 (b) 得到的结果和例 11-1 中等价的非系统前馈编码器的 $IOWEF$ 、 $CWEF$ 和 WEF 的计算结果进行比较。
- 11-22 对收尾卷积编码器的情形, 验证公式 (11-124) 给出的 $IOWEF$ 的计算结果。
- 11-23 对例 11-14, 考虑对公式 (11-140) 中的 $G(D)$ 乘以 $g^{(0)}(D) = 1 + D + D^2$ 得到的等价非系统前馈编码器。
- 1) 画出该编码器的扩充的修正状态图。
 - 2) 计算该编码器的 $IRWEF A(W, X, L)$, 三个最低输入重量 $CWEF$ 和 $WEF A(X)$ 。
 - 3) 将 (b) 得到的结果和例 11-14 中等价的非系统反馈编码器的 $IOWEF$ 、 $CWEF$ 和 WEF 的计算结果进行比较。
- 11-24 在例 11-14 中, 验证得到公式 (11-154) 中比特 WEF 的计算结果的所有步骤。
- 11-25 考虑 $G(D) = [1 \ 1 + D^2]$ 的 (2, 1, 2) 系统前馈编码器。
- 1) 画出该编码器的扩充的修正状态图。
 - 2) 计算该编码器的 $IRWEF A(W, Z, L)$, 三个最低输入重量 $CWEF$, 和 $WEF A(X)$ 。
- 11-26 使用例 11-14 的状态变量方法重新计算例 11-12 中的 $IOWEF A(W, X, L)$ 。
- 11-27 使用例 11-14 的状态变量方法重新计算例 11-13 中的 $WEF A(X)$ 。
- 11-28 考虑由习题 11-1 中的编码器生成的 (3, 1, 2) 码。
- 1) 计算自由距离 d_{free} 。
 - 2) 画出完整的 CDF。
 - 3) 计算最小距离 d_{min} 。
- 11-29 对由习题 11-15 中编码器生成的码重复习题 11-28。
- 11-30 1) 证明自由距离 d_{free} 是独立于编码器实现的, 即它是码的性质。
2) 证明 CDF 是独立于编码器实现的, 即它是码的性质。(假定 $k \times n$ 子矩阵 G_0 满秩。)
- 11-31 证明对非恶性编码器

$$\lim_{l \rightarrow \infty} d_l = d_{free}.$$

参考文献

- [1] P Elias, "Coding for Noisy Channels," *IRE Conv. Rec.*, p. 4: 37-47, 1955
- [2] J M Wozencraft and B Reiffen, *Sequential Decoding*, MIT Press, Cambridge, 1961
- [3] J L Massey, *Threshold Decoding*, MIT Press, Cambridge, 1963
- [4] A J Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Inform. Theory*, IT-13: 260-69, April 1967
- [5] L R Bahl, J Cocke, F Jelinek, and J Raviv, "Optimal Decoding of Linear Codes for Minimal Symbol Error Rate," *IEEE Trans. Inform. Theory*, IT-20: 284-87; March 1974
- [6] G Ungerboeck and I Csajka, "On Improving Data-Link Performance by Increasing the Channel Alphabet and Introducing Sequence Coding," in *IEEE International Symposium on Information Theory (ISIT1976) Book of Abstracts*, p. 53, Ronneby, Sweden, June 1976
- [7] G Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. Inform. Theory*, IT-28: 55-67, January 1982
- [8] C Berrou, A Glavieux, and P Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," in *Proc. IEEE International Conference on Communications (ICC 93)*, 1064-70, Geneva,

Switzerland, May 1993

- [9] C Berrou and A Glavieux, "Near Optimum Error Correcting and Decoding: Turbo-codes," *IEEE Trans. Commun.*, COM-44: 1261-71, October 1996
- [10] A Dholakia, *Introduction to Convolutional Codes*. Kluwer Academic, Dordrecht, The Netherlands, 1994
- [11] L H C Lee, *Convolutional Coding: Fundamentals and Applications*. Norwood, Mass., Artech House, 1997
- [12] R Johannesson and K S Zigangirov, *Fundamentals of Convolutional Coding*, IEEE Press, Piscataway, N. J., 1999
- [13] G D Forney Jr, "Convolutional Codes I: Algebraic Structure," *IEEE Trans. Inform. Theory*, IT-16: 720-38, Nov. 1970
- [14] J L Massey and M K Sain, "Inverses of Linear Sequential Circuits," *IEEE Trans. Computers*, C-17: 330-37, April 1968
- [15] S Mason and H Zimmermann, *Electronic Circuits, Signals, and Systems*. John Wiley, New York, N. Y., 1960
- [16] R Johannesson and K S Zigangirov, "Distances and Distance Bounds for Convolutional Codes," in *Topics in Coding Theory—In Honour of Lars H. Zetterberg*, G. Einarsson et al., eds., 109-36, Springer-Verlag, Berlin, 1989
- [17] G D Forney Jr, "Convolutional Codes II: Maximum Likelihood Decoding," *Inform. and Control*, 25: 222-66, July 1974
- [18] D J Costello Jr, "Free Distance Bounds for Convolutional Codes," in *IEEE Trans. Inform. Theory*, IT-20: 356-65, May 1974

