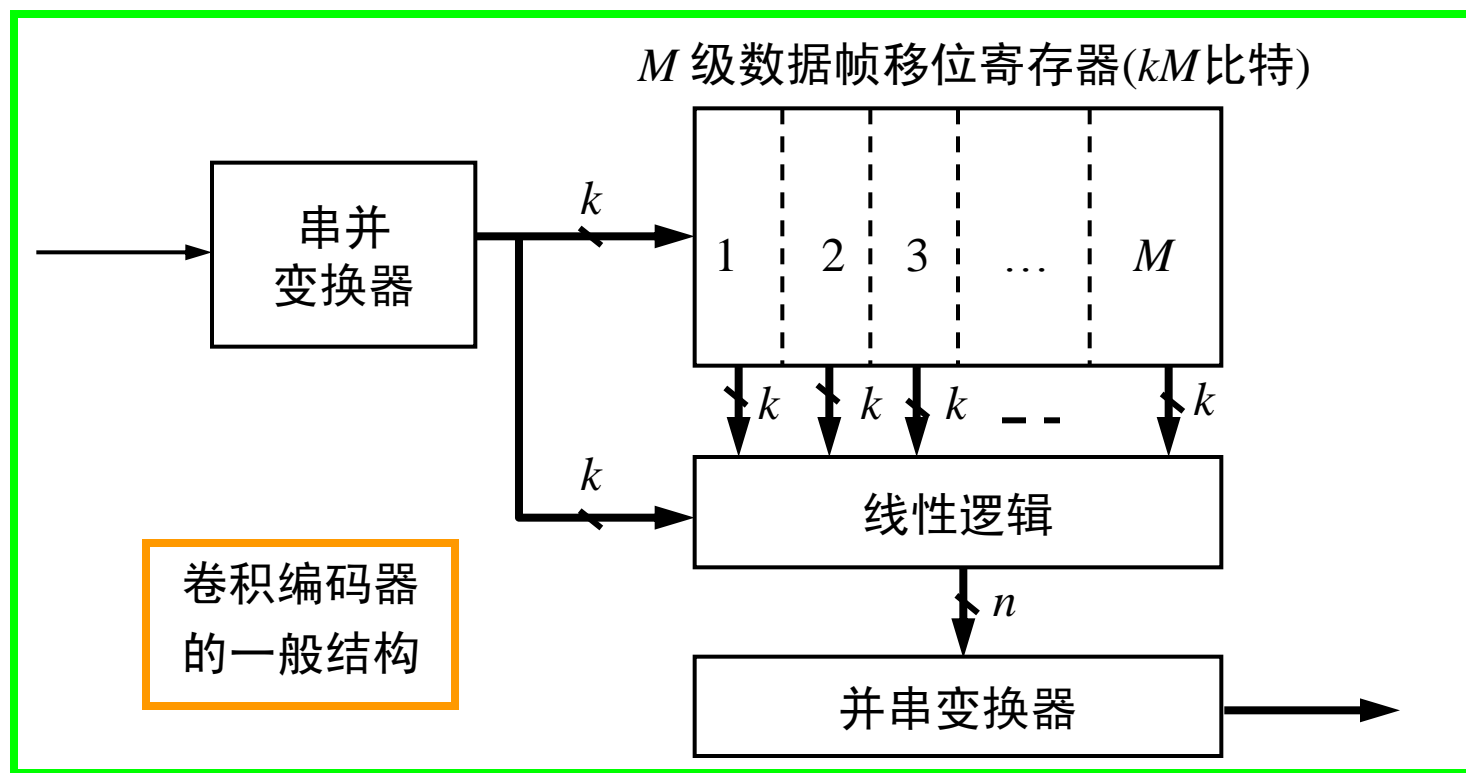


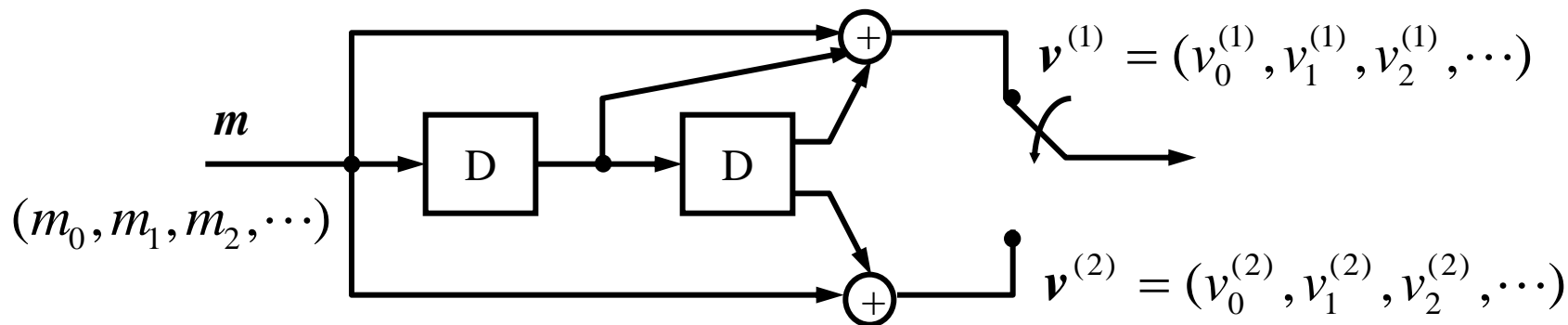
§ 9.7 卷积码的结构

卷积码的编码器是具有记忆的。在任何给定时刻编码器的 n 个输出比特不仅和当前的一组 k 个输入比特有关，而且和以前 M 个时刻的输入组有关，所以卷积码可用参数组 (n, k, M) 来描述。编码速率 $R = k/n$ 。对于卷积码来说，约束长度 $K = M + 1$ 也是一个重要参数。

9.7.1 卷积码的构成和代数描述



[例] (2, 1, 2) 卷积码编码器(R=1/2)



一、卷积码编码器的冲击响应和生成矩阵

冲激响应是当系统输入序列为 $m = (1 \ 0 \ 0 \ \cdots)$ 时，系统输出序列。

长度
3

上例中两个冲击响应为：

$$\begin{cases} g^{(1)} = (1 \ 1 \ 1) \\ g^{(2)} = (1 \ 0 \ 1) \end{cases}$$

二个输出编码序列：

$$\begin{cases} v^{(1)} = m * g^{(1)} \\ v^{(2)} = m * g^{(2)} \end{cases}$$

模2卷积运算

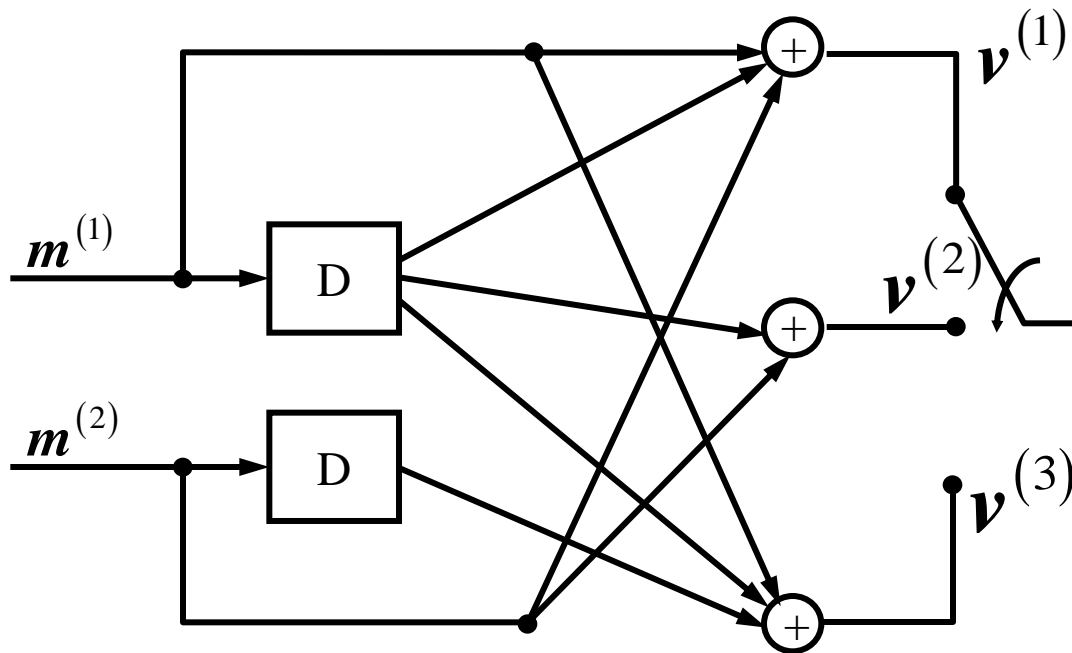
$$v_l^{(j)} = \sum_{i=0}^{\infty} m_{l-i} g_i^{(j)} = m_l g_0^{(j)} + m_{l-1} g_1^{(j)} + \cdots + m_{l-m} g_m^{(j)}, \quad i = 1, 2$$

$$G = \begin{bmatrix} g_0^{(1)} g_0^{(2)}, & g_1^{(1)} g_1^{(2)}, & g_2^{(1)} g_2^{(2)}, & g_3^{(1)} g_3^{(2)}, & \dots \\ & g_0^{(1)} g_0^{(2)}, & g_1^{(1)} g_1^{(2)}, & g_2^{(1)} g_2^{(2)}, & g_3^{(1)} g_3^{(2)}, & \dots \\ & & g_0^{(1)} g_0^{(2)}, & g_1^{(1)} g_1^{(2)}, & g_2^{(1)} g_2^{(2)}, & g_3^{(1)} g_3^{(2)}, & \dots \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & \ddots & \end{bmatrix}$$

$$= \begin{bmatrix} 11, & 10, & 11, & \dots \\ & 11, & 10, & 11, & \dots \\ & & 11, & 10, & 11, & \dots \\ & & & 11, & 10, & 11, & \dots \\ & & & & \ddots & & \\ & & & & & \ddots & \end{bmatrix}$$

编码方程： $\mathbf{v} = \mathbf{m} \cdot \mathbf{G}$

[例] (3, 2, 1) 卷积码编码器(R=2/3)



$$\mathbf{g}_1^{(1)} = (11) \quad \mathbf{g}_2^{(1)} = (10)$$

$$\mathbf{g}_1^{(2)} = (01) \quad \mathbf{g}_2^{(2)} = (10)$$

$$\mathbf{g}_1^{(3)} = (11) \quad \mathbf{g}_2^{(3)} = (01)$$

$$\begin{aligned} \mathbf{v}^{(1)} &= \mathbf{m}^{(1)} * \mathbf{g}_1^{(1)} + \mathbf{m}^{(2)} * \mathbf{g}_2^{(1)} \\ \mathbf{v}^{(2)} &= \mathbf{m}^{(1)} * \mathbf{g}_1^{(2)} + \mathbf{m}^{(2)} * \mathbf{g}_2^{(2)} \\ \mathbf{v}^{(3)} &= \mathbf{m}^{(1)} * \mathbf{g}_1^{(3)} + \mathbf{m}^{(2)} * \mathbf{g}_2^{(3)} \end{aligned}$$

$$G = \begin{bmatrix} g_{1,0}^{(1)}, g_{1,0}^{(2)}, g_{1,0}^{(3)} & g_{1,1}^{(1)}, g_{1,1}^{(2)}, g_{1,1}^{(3)} & g_{1,2}^{(1)}, g_{1,2}^{(2)}, g_{1,2}^{(3)} & \cdots & g_{1,M}^{(1)}, g_{1,M}^{(2)}, g_{1,M}^{(3)} & \cdots \\ g_{2,0}^{(1)}, g_{2,0}^{(2)}, g_{2,0}^{(3)} & g_{2,1}^{(1)}, g_{2,1}^{(2)}, g_{2,1}^{(3)} & g_{2,2}^{(1)}, g_{2,2}^{(2)}, g_{2,2}^{(3)} & \cdots & g_{2,M}^{(1)}, g_{2,M}^{(2)}, g_{2,M}^{(3)} & \cdots \\ & g_{1,0}^{(1)}, g_{1,0}^{(2)}, g_{1,0}^{(3)} & g_{1,1}^{(1)}, g_{1,1}^{(2)}, g_{1,1}^{(3)} & \cdots & g_{1,M-1}^{(1)}, g_{1,M-1}^{(2)}, g_{1,M-1}^{(3)} & \cdots \\ & g_{2,0}^{(1)}, g_{2,0}^{(2)}, g_{2,0}^{(3)} & g_{2,1}^{(1)}, g_{2,1}^{(2)}, g_{2,1}^{(3)} & \cdots & g_{2,M-1}^{(1)}, g_{2,M-1}^{(2)}, g_{2,M-1}^{(3)} & \cdots \\ & \ddots & & & & \ddots & \end{bmatrix}$$

$$\mathbf{v} = \mathbf{m} \cdot \mathbf{G}$$

$$G = \begin{bmatrix} 101 & 111 & \dots & \\ 110 & 001 & \dots & \\ & 101 & 111 & \\ & 110 & 001 & \\ & & 101 & 111 \\ & & 10 & 001 \\ & & & \ddots \end{bmatrix}$$

$$\mathbf{m} = (11, 01, 10, \dots)$$

$$\mathbf{v} = \mathbf{m} \cdot G = (011, 000, 100, 111, \dots)$$

二、卷积码编码器的多项式描述

用多项式表示输入、输出、和冲击响应序列：

$$m^{(i)}(x) = m_0^{(i)} + m_1^{(i)}x + m_2^{(i)}x^2 + \cdots, \quad i = 1, 2$$

$$v^{(j)}(x) = v_0^{(j)} + v_1^{(j)}x + v_2^{(j)}x^2 + \cdots, \quad j = 1, 2, 3$$

$$g_i^{(j)}(x) = g_{i0}^{(j)} + g_{i1}^{(j)}x + \cdots + g_{iM}^{(j)}x^M, \quad i = 1, 2 \quad j = 1, 2, 3$$

输入、输出关系：

$$v^{(1)}(x) = m^{(1)}(x) \cdot g_1^{(1)}(x) + m^{(2)}(x) \cdot g_2^{(1)}(x)$$

$$v^{(2)}(x) = m^{(1)}(x) \cdot g_1^{(2)}(x) + m^{(2)}(x) \cdot g_2^{(2)}(x)$$

$$v^{(3)}(x) = m^{(1)}(x) \cdot g_1^{(3)}(x) + m^{(2)}(x) \cdot g_2^{(3)}(x)$$

矩阵表示输入、输出关系：

$$\left(v^{(1)}(x), v^{(2)}(x), v^{(3)}(x) \right) = \left(m^{(1)}(x), m^{(2)}(x) \right) \cdot G(x)$$

$$G(x) = \begin{pmatrix} g_1^{(1)}(x) & g_1^{(2)}(x) & g_1^{(3)}(x) \\ g_2^{(1)}(x) & g_2^{(2)}(x) & g_2^{(3)}(x) \end{pmatrix}$$

9.7.2 卷积码的图描述和重量计数

一、卷积码的树图描述

寄存器中的内容表示
该时刻编码器的状态

$$S_0 = (00)$$

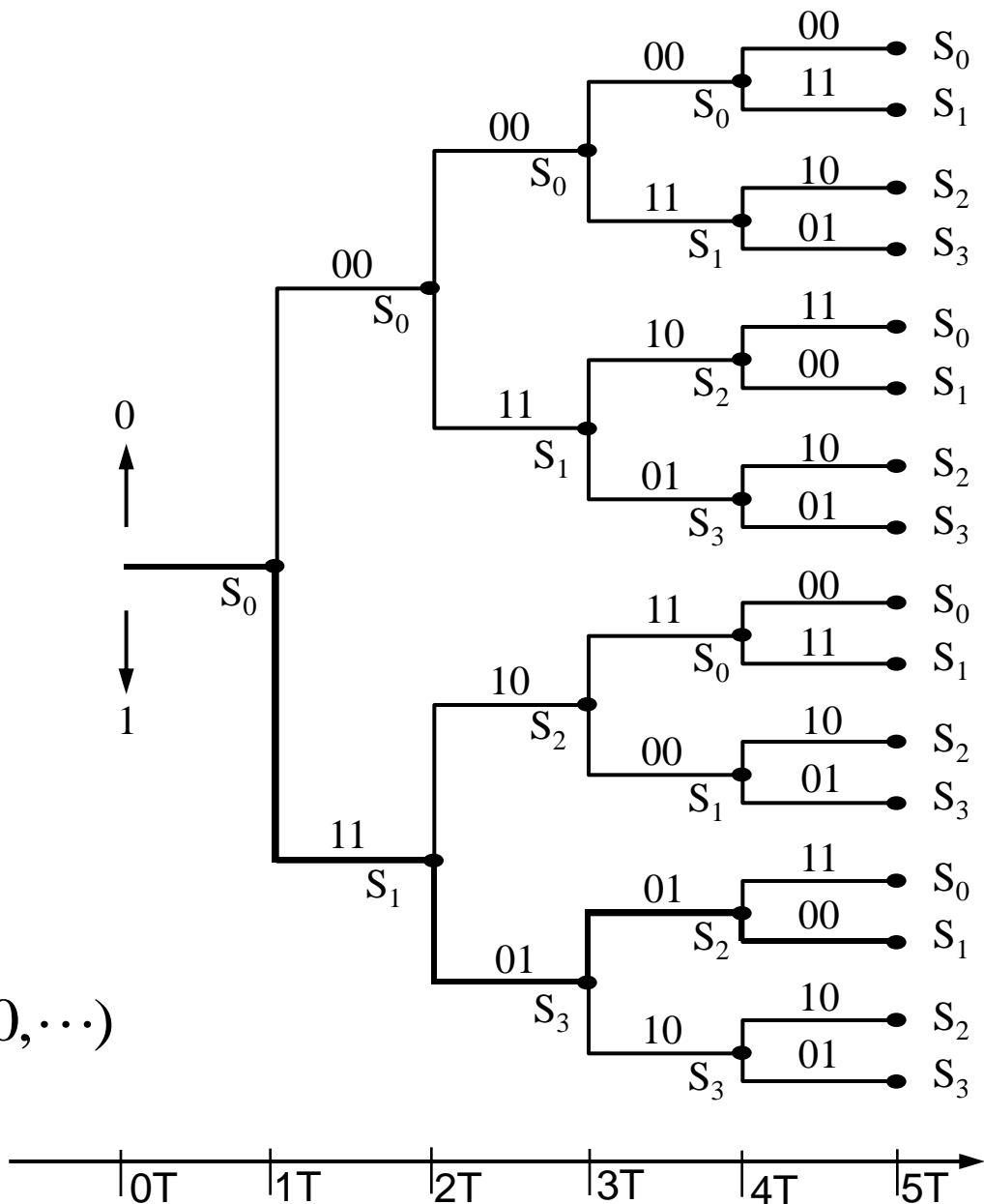
$$S_1 = (10)$$

$$S_2 = (01)$$

$$S_3 = (11)$$

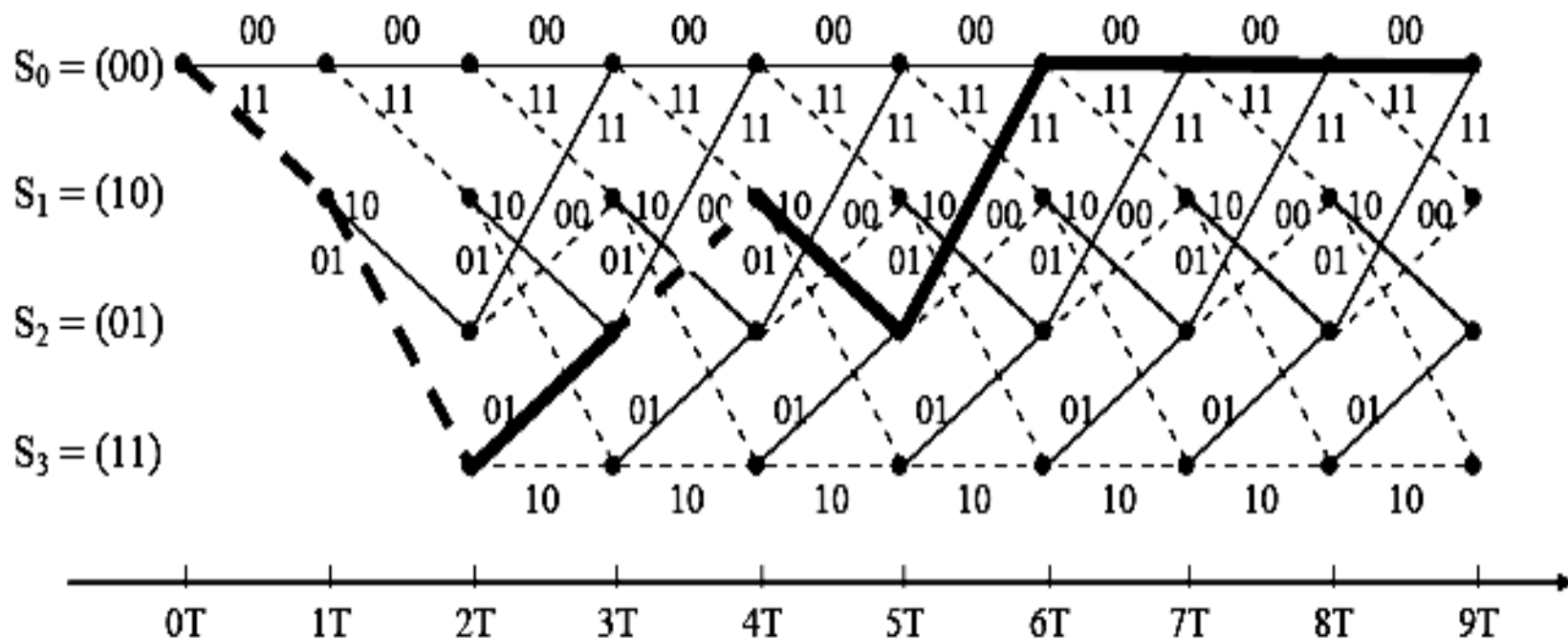
$$\mathbf{m} = (1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \dots)$$

$$\mathbf{v} = (11, 01, 01, 00, 10, 11, 00, \dots)$$



二、卷积码的网格图描述

树图随路径长度 L 的增加，终端分支数呈指数增长，故对于大的 L 不可能画出编码树。把树的每一层上同类节点归并压缩，得到网格图。



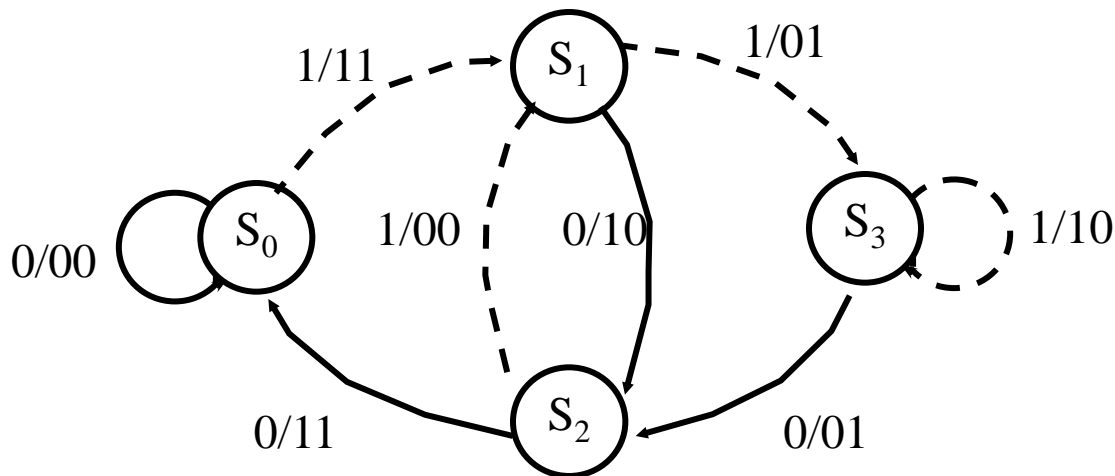
状态 S_0 出发，输入序列为： $m = (11010000\cdots)$

$$S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_2 \rightarrow S_1 \rightarrow S_2 \rightarrow S_0 \rightarrow S_0$$

$$v = (11, 01, 01, 00, 10, 11, 00, 00, 00, \cdots)$$

三、卷积码的状态图描述

卷积码编码器是一个有限状态机，因此可以用状态转移图来描述。



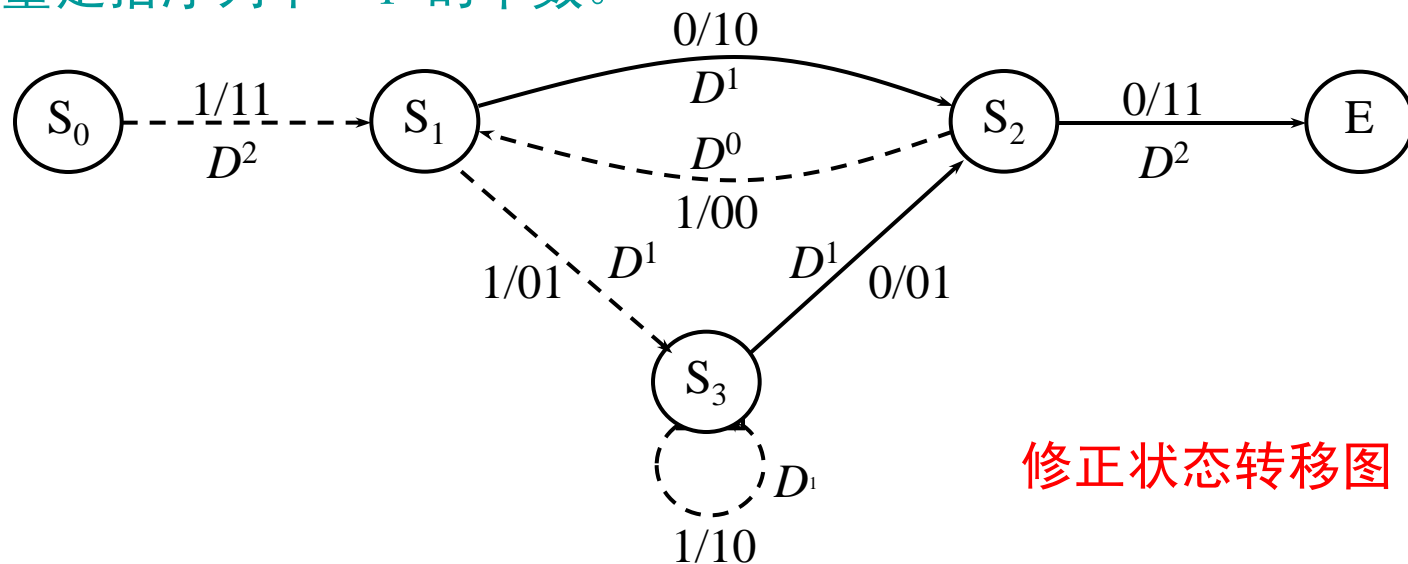
(2,1,2) 卷积码对应的状态转移图

从 S_0 状态出发，**输入** $\mathbf{m} = (01, 11, 10, 00, 10, 10)$

输出 $\mathbf{v} = (110, 010, 011, 111, 101, 010)$

9.7.3 卷积码的重量计数

线性分组码中，码字重量分布对于分组码的性能有重要的影响，对于卷积码来说输出序列的重量分布也有重要意义。所谓序列的Hamming重量是指序列中“1”的个数。



修正状态转移图

- 1、连结 S_0 和 E 的每一条路径都是从 S_0 状态出发又首次回到 S_0 状态的路径，它代表一个非零码字。
- 2、每条路径的总增益等于沿此路径的各分支增益之积，相应的码字重量等于路径增益中 D 的幂次。

用状态变量 Z_0, Z_1, Z_2, Z_3, Z_E 分别表示从 S_0 出发, 终止于 S_0, S_1, S_2, S_3 和 E 的所有路径增益和。

$$Z_1 = D^2 \cdot Z_0 + Z_2$$

$$Z_2 = D \cdot Z_1 + D \cdot Z_3$$

$$Z_3 = D \cdot Z_1 + DZ_3$$

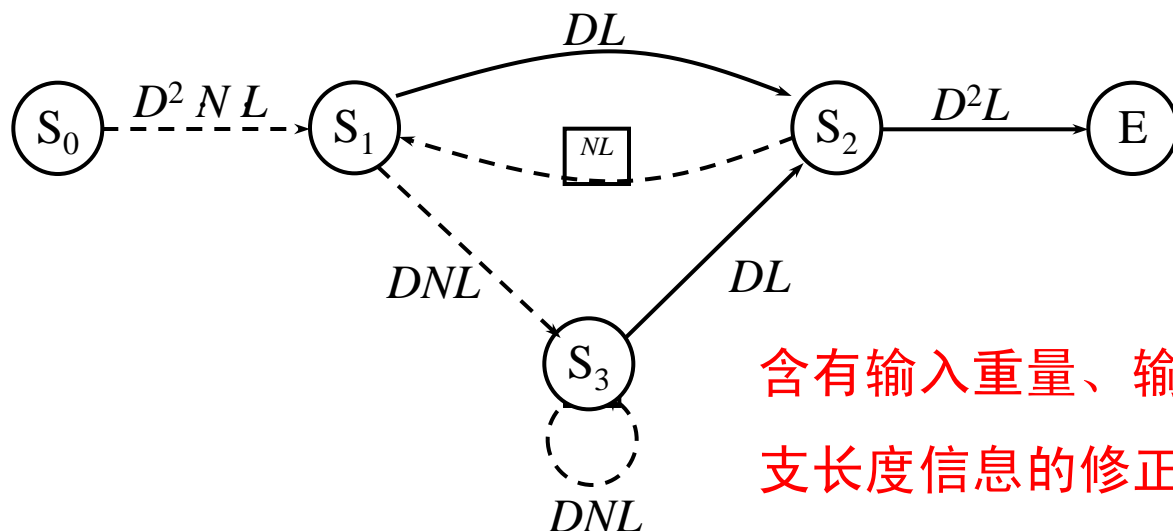
$$Z_E = D^2 \cdot Z_2$$

$$\begin{aligned} T(D) &= \frac{Z_E}{Z_0} = \frac{D^5}{1-D} \\ &= D^5 (1 + 2D + 4D^2 + 8D^3 + \cdots + 2^l D^l + \cdots) \\ &= 1 \cdot D^5 + 2 \cdot D^6 + 4 \cdot D^7 + 8 \cdot D^8 + \cdots + 2^l \cdot D^{l+5} + \cdots \end{aligned}$$

从重量分布公式可见, 重量为 $(l+5)$ 的码字有 2^l 个。

在分支增益上添加其它的因子，还能获得非零码字的其它结构信息。如分支增益中因子 N^j 的指数 j 表示相应输入 k 个比特消息的重量（输入数据中“1”的个数），另外每个分支增益中增加一个因子 L ，表示一个分支

长度。



含有输入重量、输出重量和分支长度信息的修正状态转移图

$$Z_1 = D^2 \cdot NLZ_0 + NLZ_2$$

$$Z_2 = DLZ_1 + DLZ_3$$

$$Z_3 = DNLZ_1 + DNLZ_3$$

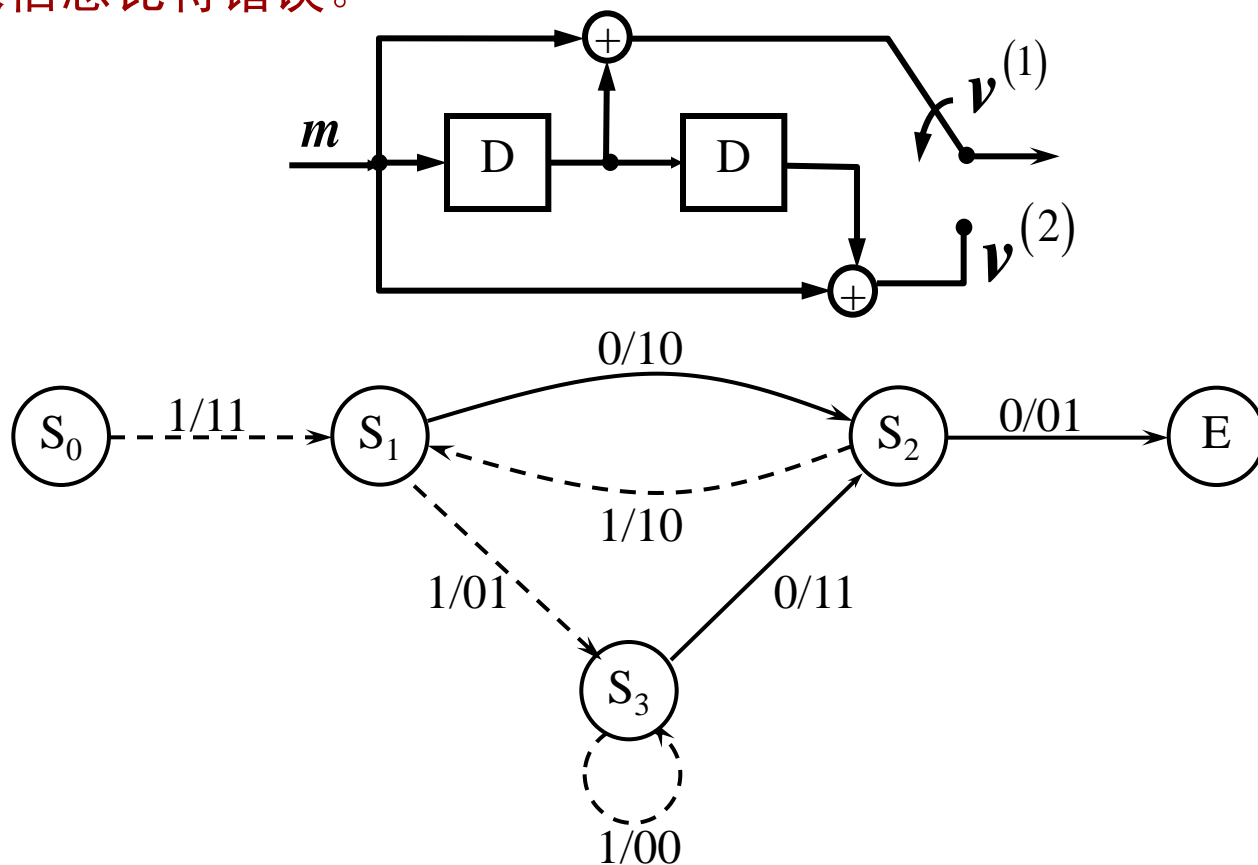
$$Z_E = D^2 LZ_2$$

$$T(D, L, N) = Z_E / Z_1 = \frac{D^5 L^3 N}{1 - DL(1 + L)N}$$

$$= D^5 L^3 N + D^6 L^4 (1 + L) N^2 + D^7 L^5 (1 + L)^2 N^3 + \dots + D^{l+5} L^{l+3} (1 + L)^l N^{l+1} + \dots$$

9.7.4 恶性码

卷积码是恶性码的充要条件是状态图中存在一个输出重量为零的回路，但是 S_0 上的自环除外。恶性码使得有限多个码字比特错误造成无限多个输入信息比特错误。



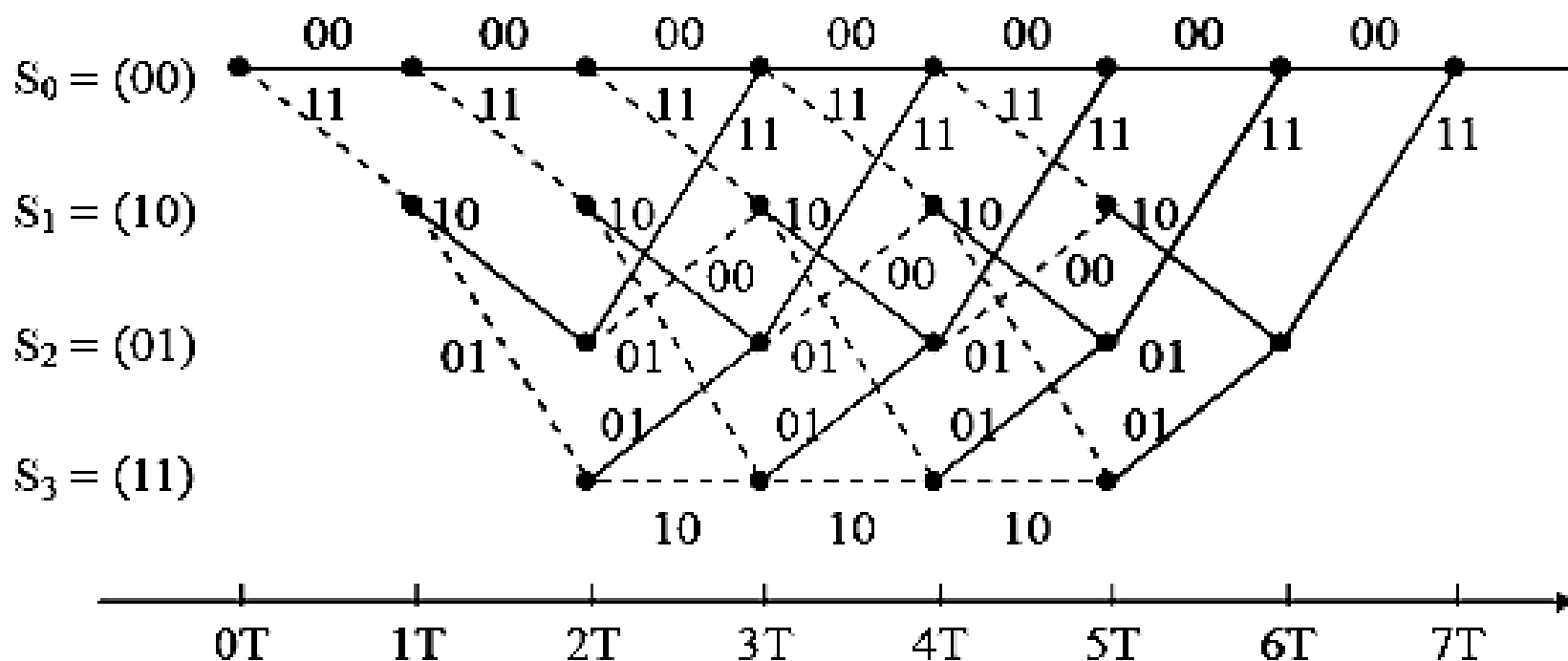
恶性码例子

§ 9.8 卷积码的Viterbi译码算法

Viterbi算法等价于在加权图上求最短路径；Viterbi算法是卷积码的最大似然译码算法。

考虑图9.7.2所示的(2,1,2)卷积码，它的生成多项式矩阵为，

$$G(x) = (1 + x + x^2, 1 + x^2)$$



编码器状态从 S_0 起始，并回到 S_0 。前面 M 个时刻，对应于起始阶段；而最后 M 个时刻，通过输入 M 个“0”，使译码器返回 S_0 状态。

长度为 kL 的消息序列： $\mathbf{m} = (m_0, m_1, \dots, m_{L-1})$ $m_i \in \{0, 1\}^k$

相应的码字序列： $\mathbf{v} = (v_0, v_1, \dots, v_{L+M-1})$ $v_j \in \{0, 1\}^n$

接收到序列： $\mathbf{r} = (r_0, r_1, \dots, r_{L+M-1})$ $r_j \in Y^n$

二元硬判决信道： $Y = \{0, 1\}$

高斯信道： $Y = R$

在接收到 \mathbf{r} 时，发送序列 \mathbf{v} 的似然函数为：

$$P(\mathbf{r} | \mathbf{v}) = \prod_{i=0}^{L+M-1} P(r_i | v_i) \iff \log P(\mathbf{r} | \mathbf{v}) = \sum_{i=0}^{L+M-1} \log P(r_i | v_i)$$

最大似然估计码字序列 $\hat{\mathbf{v}}$ 为

$$\hat{\mathbf{v}} = \arg \max_{\mathbf{v} \in V} \log P(\mathbf{r} | \mathbf{v})$$

路径 \mathbf{v} 的度量: $\lambda(\mathbf{r} | \mathbf{v}) \triangleq \log P(\mathbf{r} | \mathbf{v}) = \sum_{i=0}^{L+M-1} \log P(r_i | v_i)$

$$= \sum_{i=0}^{L+M-1} \lambda(r_i | v_i)$$

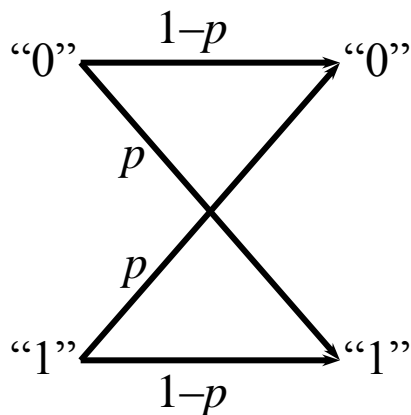
分支度量: $\lambda(r_i | v_i) = \log P(r_i | v_i)$

所以一条路径的度量为该路径上各分支度量之和。

一条路径前 l 个分支所构成的部分路径度量表示为:

$$\lambda((\mathbf{r} | \mathbf{v})_0^{l-1}) = \sum_{i=0}^{l-1} \lambda(r_i | v_i)$$

1、硬判决信道



$$P(r_i | v_i) = p^{d_i} (1-p)^{n-d_i}$$

$$\lambda(r_i | v_i) = d_i \cdot \log \frac{p}{1-p} + n \log(1-p) \triangleq \alpha \cdot d_i + \beta$$

$$\lambda(\mathbf{r} | \mathbf{v}) = \sum_{i=0}^{L+M-1} \lambda(r_i | v_i) \triangleq \alpha \sum_{i=0}^{L+M-1} d_i + \beta(L+M)$$

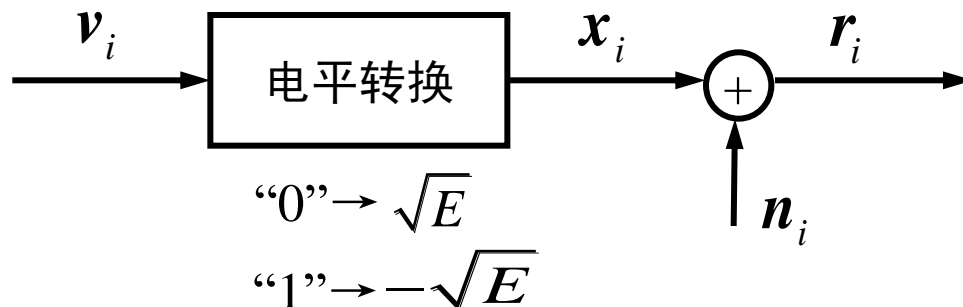
最大似然译码



最小Hamming距离译码

$P < 0.5$

2、高斯信道



$$\mathbf{r}_i = \mathbf{x}_i + \mathbf{n}_i = (x_{i1} + n_{i1}, x_{i2} + n_{i2}, \dots, x_{in} + n_{in})$$

$$P(\mathbf{r}_i | \mathbf{v}_i) = P(\mathbf{r}_i | \mathbf{x}_i) = \prod_{j=1}^n \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left\{-\frac{(r_{ij} - x_{ij}\sqrt{E})^2}{2\sigma_n^2}\right\}$$

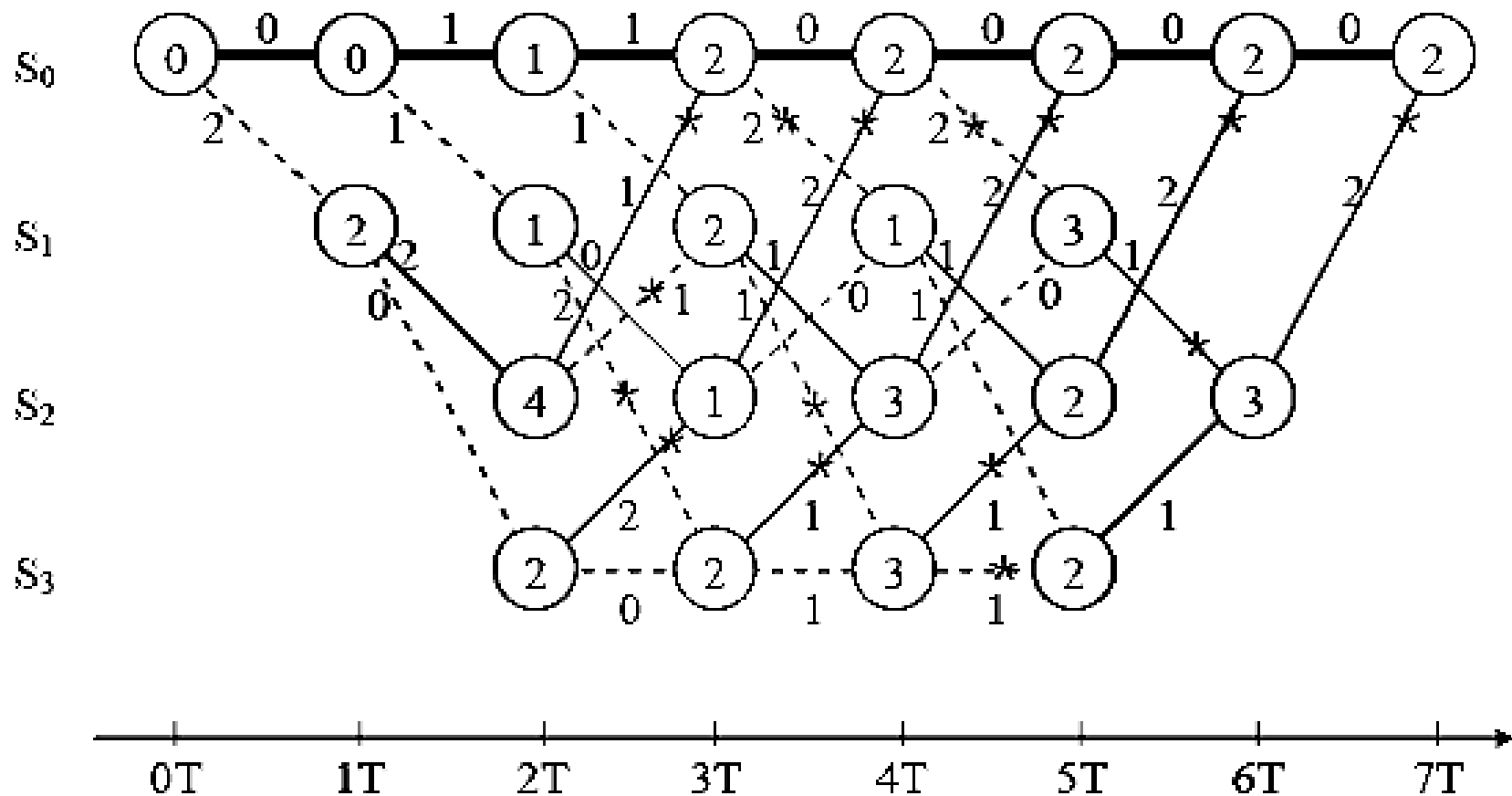
$$\lambda(\mathbf{r}_i | \mathbf{v}_i) = \log P(\mathbf{r}_i | \mathbf{v}_i) = C \sum_{j=1}^n r_{ij} \cdot x_{ij} + D$$

$C > 0$,
 D 常数

$$\lambda(\mathbf{r} | \mathbf{v}) = C \sum_{i=0}^{L+M-1} \sum_{j=1}^n r_{ij} \cdot x_{ij} + D(L+M)$$

最大似然译码 \longleftrightarrow 选互相关最大的路径

9.8.2 Viterbi译码算法



接收序列: $\mathbf{r} = (00, 01, 10, 00, 00, 00, 00)$

最后幸存路径: $S_0 \rightarrow S_0 \rightarrow S_0 \rightarrow S_0 \rightarrow S_0 \rightarrow S_0 \rightarrow S_0$

判定发送序列: $\mathbf{m} = (0, 0, 0, 0, 0, 0, 0)$

9.8.3 作为前向动态规划解的Viterbi算法

以 $(n, 1, M)$ 卷积码为例, 说明Viterbi算法是在加权网格图上寻找最大路径值的前向动态规划解。

卷积码编码器输入序列:

$$\mathbf{m} = (m_0, m_1, m_2, \cdots) \quad m_i \in \{0, 1\}$$

编码器具有 M 个寄存器, 在 $t = kT$ 时刻编码器状态为:

$$\sigma_k = (m_{k-1}, m_{k-2}, \cdots, m_{k-M}) \in \{0, 1\}^M$$

在 kT 时刻编码器输出码字:

$$\mathbf{v}_k = f(m_k, \sigma_k) = f(m_k, m_{k-1}, \cdots, m_{k-M})$$

状态转移: $\sigma_k \xrightarrow{m_k} \sigma_{k+1}$

$$\begin{aligned} \lambda(\mathbf{r} | \mathbf{v}) &= \sum_{i=0}^{L+M-1} \lambda(\mathbf{r}_i | \mathbf{v}_i) \\ &= \sum_{i=0}^{L+M-1} \lambda[\mathbf{r}_i | f(m_i, m_{i-1}, \cdots, m_{i-M})] \end{aligned}$$

限定路径起始于全零状态，最后终止于全零状态，即

$$m_{-M} = m_{-M+1} = \cdots = m_{-1} = 0$$

$$m_L = m_{L+1} = \cdots = m_{L+M-1} = 0$$

最大似然译码就是在网络图上寻找一条满足初始和终止条件的路径

$$(\underbrace{0, 0, \cdots, 0}_M, \underbrace{m_0, m_1, \cdots, m_{L-1}}_L, \underbrace{0, \cdots, 0}_M)$$

使得路径度量值为最大，即

$$J = \max_{\{m_j\}_0^{L-1}} \sum_{i=0}^{L+M-1} \lambda[\mathbf{r}_i \mid f(m_i, m_{i-1}, \cdots, m_{i-M})]$$

零时刻从全零状态出发，在第 k 时刻到达状态 σ_k 的幸存路径所对应的

部分路径度量值定义为：

$$\begin{aligned} J_k(\sigma_k) &= J_k(m_{k-1}, m_{k-2}, \cdots, m_{k-M}) \\ &= \max_{\{m_j\}_0^{k-M-1}} \left\{ \sum_{i=0}^{k-1} \lambda[\mathbf{r}_i \mid f(m_i, m_{i-1}, \cdots, m_{i-M})] \right\} \end{aligned}$$

递归计算:

$$\begin{aligned} J_{k+1}(\sigma_{k+1}) &= J_{k+1}(m_k, m_{k-1}, \dots, m_{k-M+1}) \\ &= \max_{\{m_j\}_0^{k-M}} \left\{ \sum_{i=0}^k \lambda[\mathbf{r}_i | f(m_i, m_{i-1}, \dots, m_{i-M})] \right\} \\ &= \max_{m_{k-M}} \max_{\{m_j\}_0^{k-M-1}} \left\{ \sum_{i=0}^{k-1} \lambda[\mathbf{r}_i | f(m_i, m_{i-1}, \dots, m_{i-M})] + \right. \\ &\quad \left. \lambda[\mathbf{r}_k | f(m_k, m_{k-1}, \dots, m_{k-M})] \right\} \\ &= \max_{m_{k-M}} \left\{ J_k(m_{k-1}, m_{k-2}, \dots, m_{k-M}) + \lambda[\mathbf{r}_k | f(m_k, \dots, m_{k-M})] \right\} \\ &= \max \begin{cases} J_k(m_{k-1}, \dots, m_{k-M+1}, 0) + \lambda[\mathbf{r}_k | f(m_k, \dots, m_{k-M+1}, 0)] \\ J_k(m_{k-1}, \dots, m_{k-M+1}, 1) + \lambda[\mathbf{r}_k | f(m_k, \dots, m_{k-M+1}, 1)] \end{cases} \end{aligned}$$

初始条件为: $J_0(\sigma_0 = \mathbf{0}) = J_0[\sigma_0 = (0, 0, \dots, 0)] = 0$

$$J_0(\sigma_0 \neq \mathbf{0}) = -\infty$$

对于二元对称信道: $\max \rightarrow \min$, $-\infty \rightarrow \infty$, $\lambda(\mathbf{r}_i, \mathbf{v}_i) = d_H(\mathbf{r}_i, \mathbf{v}_i)$

9.8.4 实现Viterbi译码算法的一些具体考虑

一、译码器存贮器数目

在Viterbi译码中，对每个状态必须提供存贮器来寄存幸存路径及其度量。状态数随 M 指数地增长，一般 M 取10左右。

二、路径存贮的截断

截短译码器的路径存贮：对每条幸存路径只寄存其最近 τ 个消息数据，其中 $\tau \ll L$ 。译码器处理了接收序列的前 τ 组数据后，译码存贮器就满了，必须作出强制性判决，确定第一个消息数据比特，并作为译码器最终判决输出；然后删除存贮器中第一个消息数据比特，腾出空间以暂存新来的幸存数据比特。

在任何时刻 k ($k \geq \tau$)，强制性判决可以有下面 3种可能方式：

- 1、在 2^M 条幸存路径中，任选一条，并把该路径中第 $(k - \tau)$ 时刻（即回退 τ 时刻）的消息数据为译码输出比特；
 - 2、在 2^M 个可能的第 $(k - \tau)$ 时刻消息数据中选一个出现次数最多的数据为译码输出比特；
 - 3、在 k 时刻的 2^M 条幸存路径中，具有最大部分路径度量的那一条的第 $(k - \tau)$ 时刻消息数据作为译码输出比特；
- 一般 τ 是编码器存贮器数目 M 的5倍左右。

三、译码器的同步

状态同步：

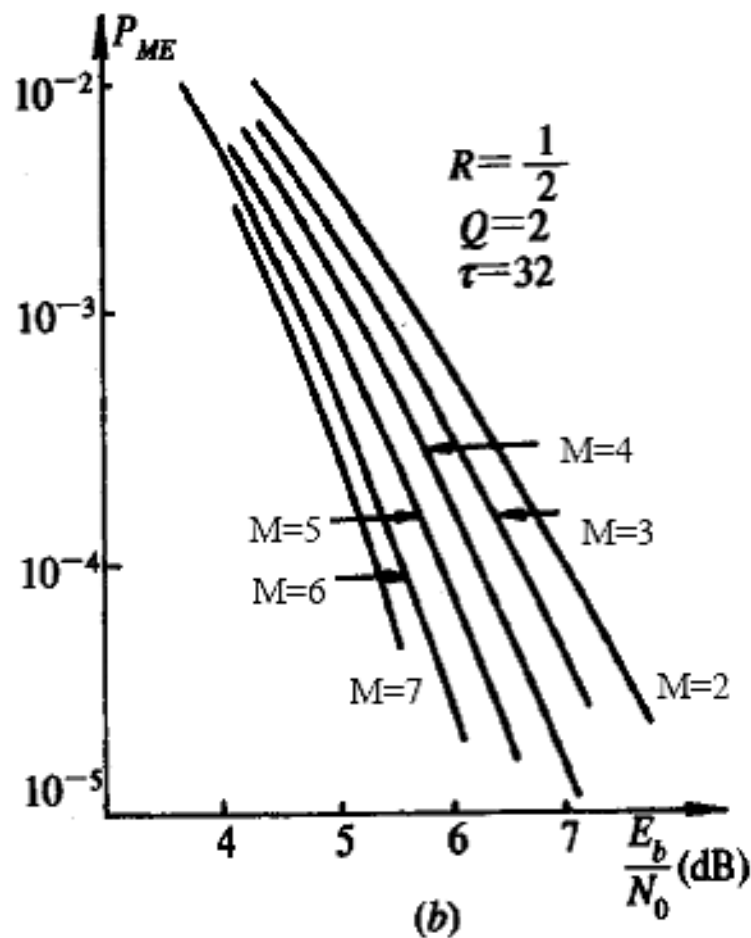
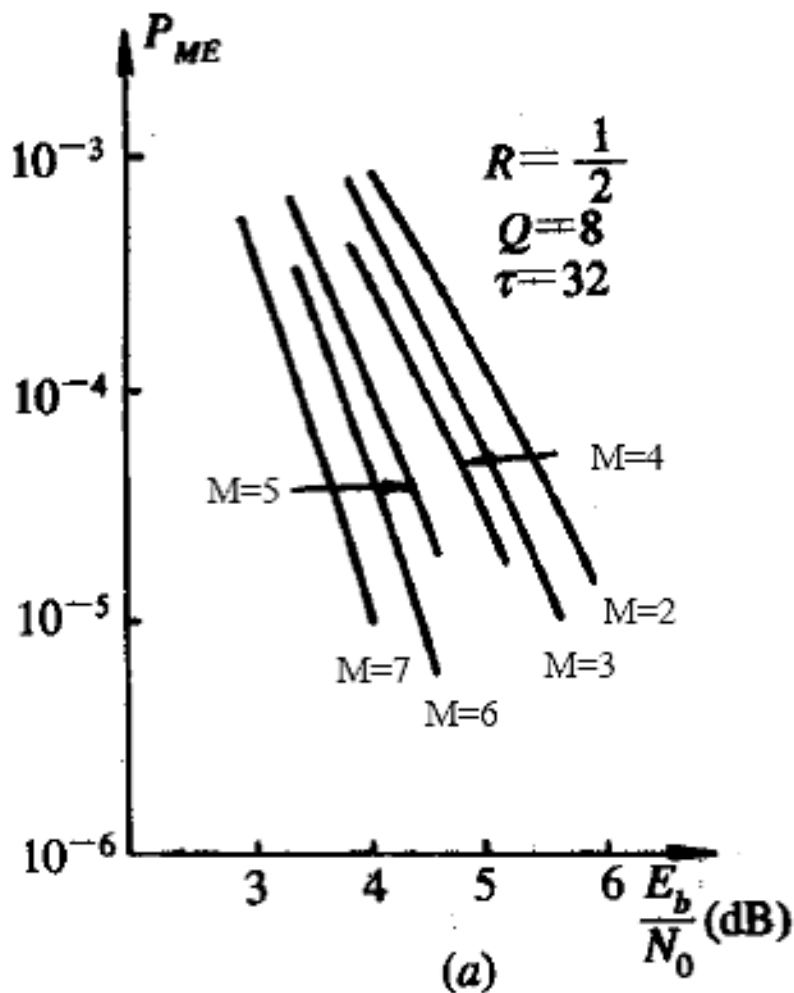
译码器可能从一个未知的编码状态开始工作，或者说译码器在开始工作时，可能处在任何一个状态中；译码器的所有状态寄存器，必须都初始化为零；Forny证明大约经过 $5M$ 个分支后这些幸存路径就会汇合。

比特同步：

位同步错误，则所有幸存路径的度量没有明显差别，以此作同步识别；

四、分支度量的量化精度

在软判决信道上卷积码译码比硬判决信道具有性能上优越，一般信道输出量化电平数越多，性能越好；但8电平（ $Q=8$ ）量化所得的性能和无量化理想情况下性能仅相差无几(0.25dB)。

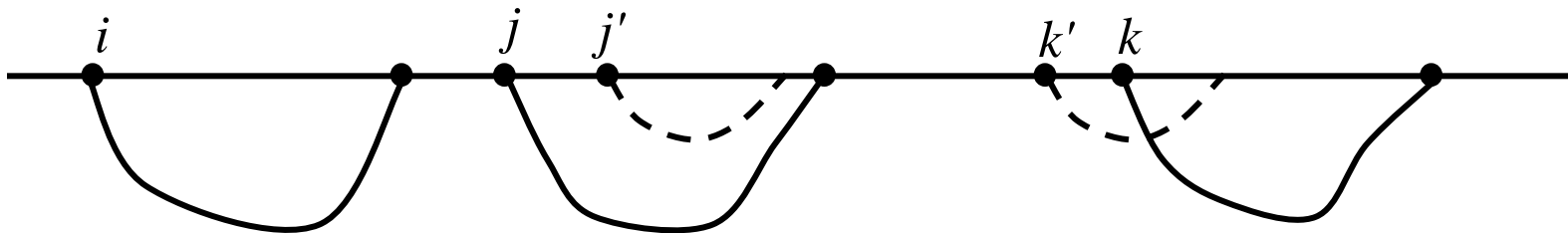


9.8.4 卷积码Viterbi译码算法的性能界

$$P_b = \frac{\text{译码输出序列中平均错误比特数}}{\text{总的传输比特数}}$$

一、节点错误概率

图中横贯全零状态的全零路，代表正确路；下面实线通路是由Viterbi算法所选择的，译码器输出路径。它和正确路有不重合的地方，表明出了错误。按Viterbi算法，在这些不重合的路径片段上正确路所积累的路径度量小于不正确路径的路径度量积累。我们把这种错误称为发生在节点 i ， j 和 k 上的节点错误。在节点 j 出现节点错误的必要，但并非充分的条件是从该节点起有一条与正确路开始分离的不正确路径在分离跨度上积累了更大的路径度量。



节点 j 上的节点错误概率

$$P_e(j) \leq P_r \left\{ \bigcup_{\mathbf{v}' \in V'(j)} [\lambda(\mathbf{r} | \mathbf{v}') > \lambda(\mathbf{r} | \mathbf{v})] \right\}$$

$V'(j)$ 表示在 j 节点与正确路分离，而后再汇合的全部不正确路径的集合。

由联合界：

$$P_e(j) \leq \sum_{\mathbf{v}' \in V'(j)} P_r \{ \lambda(\mathbf{r} | \mathbf{v}') > \lambda(\mathbf{r} | \mathbf{v}) \}$$

可以证明成对错误概率为：

$$P_r \{ \lambda(\mathbf{r} | \mathbf{v}') > \lambda(\mathbf{r} | \mathbf{v}) \} = \left\{ \sum_r \sqrt{P_0(r)P_1(r)} \right\}^d \square Z^d$$

$$Z \square \sum_r \sqrt{P_0(r)P_1(r)}, \quad P_0(r) \square P(r | v = 0), \quad P_1(r) \square P(r | v = 1)$$

$$d \square d_H(\mathbf{v}, \mathbf{v}') = w_H(\mathbf{v}')$$

所以

$$\begin{aligned} P_e(j) &\leq \sum_{d=d_{free}}^{\infty} \sum_{\mathbf{v}' \in B_d(j)} P_r \{ \lambda(\mathbf{r} | \mathbf{v}') > \lambda(\mathbf{r} | \mathbf{v}) \} \\ &\leq \sum_{d=d_{free}}^{\infty} A(d) Z^d \end{aligned}$$

其中 $B_d(j)$ 表示 $V'(j)$ 中与正确路径距离为 d 的错误路径所组成的子集合，也就是 $V'(j)$ 中重量为 d 的不正确路径组成的子集； $A(d)$ 表示 $B_d(j)$ 中元素数目。 d_{free} 称为自由距离，它表示从全零状态出发又首次回到全零状态的所有路径中的最小重量。

$$P_e(j) \leq \sum_{d=d_{free}} A(d) Z^d = T(D) \Big|_{D=Z}$$

其中 $T(D)$ 是卷积码的重量生成函数。所以节点错误概率为

$$Z^{d_{free}} \leq P_e \leq T(D) \Big|_{D=Z}$$

[例9.8.2] 对于图9.1.2所示(2, 1, 2)卷积码，重量生成函数

$$T(D) = \frac{D^5}{1-D} = D^5 + 2D^6 + 4D^7 + \cdots + 2^l \cdot D^{l+5} + \cdots$$

$$d_{free} = 5$$

$$Z^5 \leq P_e \leq \frac{D^5}{1-D} \Big|_{D=Z}$$

二、比特错误概率

令 $B_d(j, i)$ 表示 $B_d(j)$ 中由重量为 i 的输入消息序列所引起的不正确路所组成的子集。

$$B_d(j) = \bigcup_{i=1}^{\infty} B_d(j, i)$$

由节点错误引起的平均错误比特数为：

$$\begin{aligned} E[n_b(j)] &\leq \sum_{i=1}^{\infty} \sum_{d=d_{free}}^{\infty} \sum_{\mathbf{v}' \in B_d(j, i)} i \cdot P_r\{\lambda(\mathbf{r} | \mathbf{v}') > \lambda(\mathbf{r} | \mathbf{v})\} \\ &\leq \sum_{i=1}^{\infty} \sum_{d=d_{free}}^{\infty} i \cdot A(d, i) Z^d \end{aligned}$$

$A(d, i)$ 表示 $B_d(j, i)$ 中元素数目，等于 $T(D, N) = T(D, L, N)|_{L=1}$ 中 $D^d N^i$ 的系数，所以

$$E[n_b(j)] \leq \left. \frac{\partial T(D, N)}{\partial N} \right|_{N=1, D=Z}$$

$$P_b \leq \frac{E[n_b(j)]}{k} \leq \frac{1}{k} \cdot \left. \frac{\partial T(D, N)}{\partial N} \right|_{N=1, D=Z}$$

[例9.8.3] 对于例9.8.2的考虑的(2, 1, 2)卷积编码器，其生成函数为

$$T(D, N) = T(D, L, N) \Big|_{L=1} = \frac{D^5 N}{1 - 2DN}$$

$$\frac{\partial T(D, N)}{\partial N} \Big|_{N=1} = D^5 + 2 \cdot 2D^6 + \cdots + (k+1) \cdot 2^k D^{k+5} + \cdots$$

$$P_b \leq 1 \cdot Z^5 + 2 \cdot 2Z^6 + 3 \cdot 4 \cdot Z^7 + \cdots + (k+1) \cdot 2^k Z^{k+5} + \cdots$$

↑
1 比特错

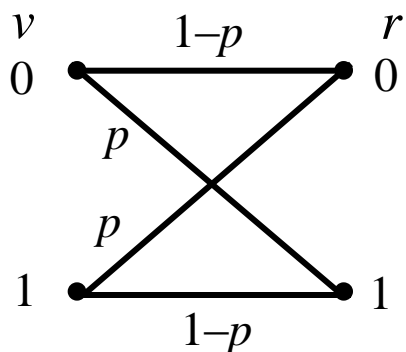
↑
2 比特错

↑
3 比特错

↑
k 比特错

9.8.5 卷积码在BSC和AWGN信道的性能

一、卷积码在BSC信道上的性能



$$Z = \sum_r \sqrt{P_0(r)P_1(r)} = 2\sqrt{p(1-p)}$$

BPSK传输差错概率:

$$p = Q\left(\sqrt{\frac{2E}{N_0}}\right) \leq \frac{1}{2} e^{-\frac{E}{N_0}} \rightarrow p \leq \frac{1}{2} e^{-RE_b / N_0}$$

$$R = k/n$$

$$E = RE_b$$

所以 $Z \leq \sqrt{2} \cdot e^{-RE_b / 2N_0}$

$$P_{b,BSC} = \frac{1}{k} \left. \frac{\partial T(D, N)}{\partial N} \right|_{N=1, D=\sqrt{2} \cdot e^{-kE_b / 2N_0}}$$

E_b/N_0
充分大

$$P_{b,BSC} \approx \frac{1}{k} \cdot B_{free} \cdot 2^{d_{free}/2} \cdot e^{-Rd_{free}E_b / 2N_0}$$

编码增益: $G = Rd_{free} / 2$

二、卷积码在AWGN信道上的性能

采用BPSK传输，接收到信号

$$r = \begin{cases} \sqrt{E} + n & \text{发送码字符号 } v = "0" \\ -\sqrt{E} + n & \text{发送码字符号 } v = "1" \end{cases}$$

$$R = k/n \\ E = RE_b$$

$$p_0(r) \sim \mathcal{N}(\sqrt{E}, N_0/2) \quad p_1(r) \sim \mathcal{N}(-\sqrt{E}, N_0/2)$$

$$Z = \sum_r \sqrt{p_0(r)p_1(r)} = e^{-E/N_0} = e^{-RE_b/N_0}$$

E_b/N_0
充分大

$$P_{b,AWGN} \leq \frac{1}{k} \frac{\partial T(D, N)}{\partial N} \bigg|_{N=1, D=e^{-kE_b/N_0}}$$

$$P_{b,AWGN} \approx \frac{1}{k} \cdot B_{d_{free}} \cdot e^{-Rd_{free} \cdot E_b/N_0}$$

编码增益:

$$G = Rd_{free}$$

三、编码增益

无编码时的误比率: $P_{b, \text{无编码}} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \leq \frac{1}{2} e^{-E_b / N_0}$

编码增益:
$$G = \frac{(E_b / N_0)_{\text{编码}}}{(E_b / N_0)_{\text{无编码}}}$$
$$= \begin{cases} Rd_{\text{free}} / 2 & \text{对BSC硬判决} \\ Rd_{\text{free}} & \text{对AWGN软判决} \end{cases}$$

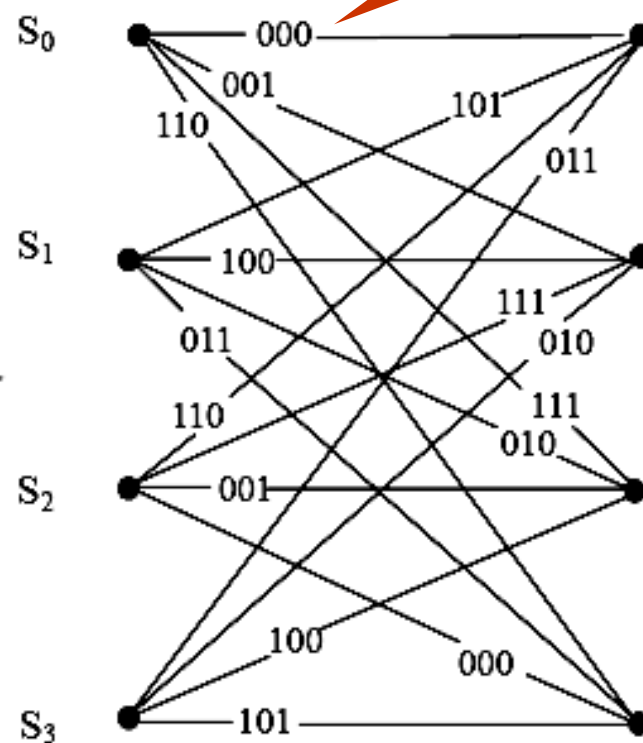
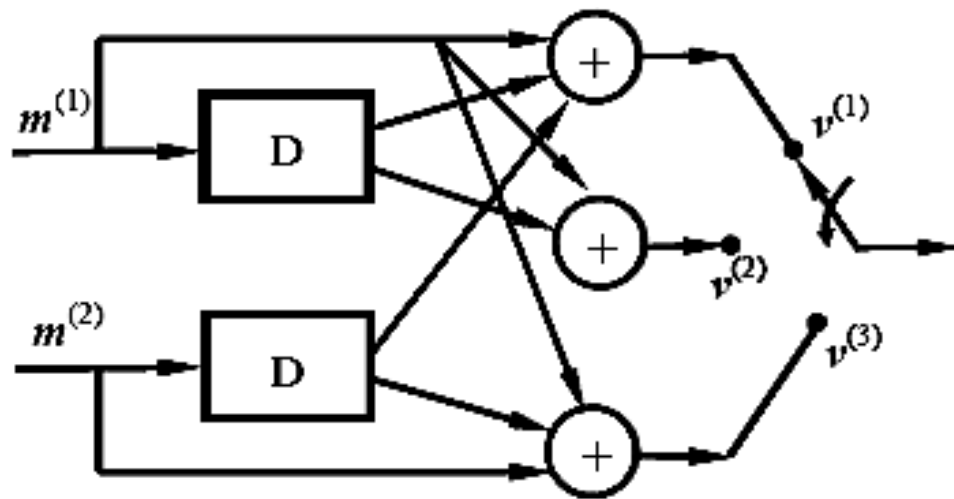
$$\frac{R \cdot d_{\text{free}}}{2} \leq (\text{码率为} R, \text{自由距离为 } d \text{ 的卷积码编码增益}) \leq Rd_{\text{free}}$$

§ 9.9 凿孔卷积码

凿孔卷积码的目的是提高卷积码的码率。

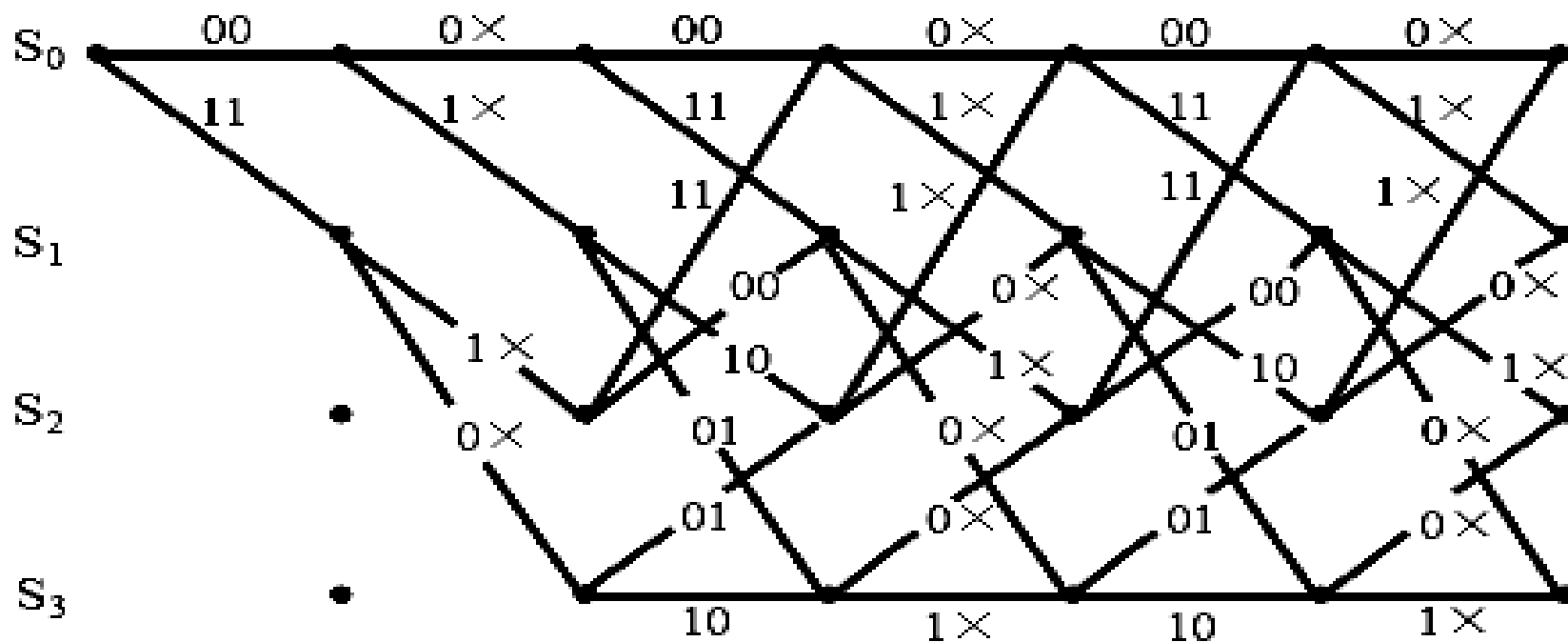
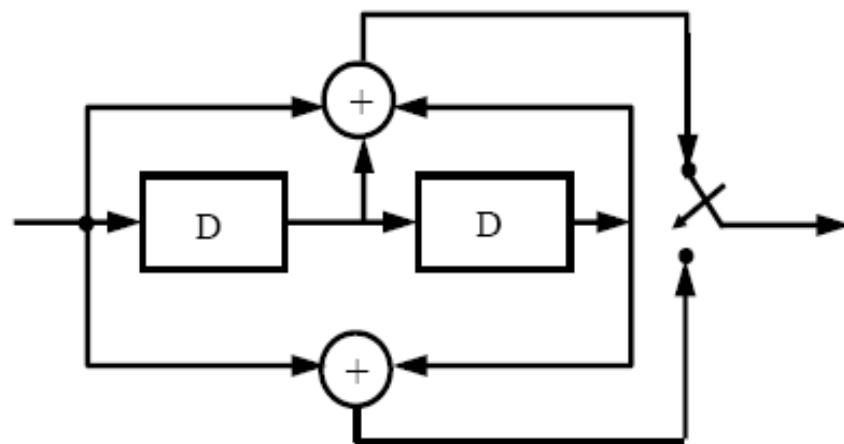
考虑卷积码，码率为 $2/3$ ，记忆 $M=1$ ，生成矩阵为

$$G(X) = \begin{pmatrix} 1+X & 1+X & 1 \\ X & 0 & 1+X \end{pmatrix}$$



复杂的
Viterbi译码

考虑码率 $R=1/2$, $M=2$, 生成多项式为 $G(X) = (1 + X + X^2, 1 + X^2)$ 的卷积码凿空, 卷积编码器每4位输出比特周期地删除掉1位 (即凿掉一位), 码率上升到 $R=3/4$ 。



凿孔后的卷积码的自由距离减小，从5降到3，但是这个自由距离不比任何其它 $R=3/4$ ，状态数为 4 的卷积码自由距离低，因而它们的性能也差不多，但译码显然容易多了。

对于卷积码进行周期为 P 的凿孔由凿孔表A确定。A是 n 行 P 列的阵列

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1P} \\ a_{21} & a_{22} & \cdots & a_{2P} \\ \cdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nP} \end{pmatrix} \quad a_{ji} \in \{0,1\}$$

$a_{ji} = 0$ 表示卷积码第 j 路输出在时刻 $(kP+i)T, k=0,1,2,\cdots$ ，周期地被删除； $a_{ji} = 1$ 表示相应输出被保留。如果A表中有 N 个1 ($N \geq P$)，则

码率：

$$R = P / N$$

分支度量：

$$\lambda(\mathbf{r}_i | \mathbf{u}_i) = \sum_{j=1}^n r_{ij} \cdot x_{ij} \cdot a_{ji}$$

码率兼容凿孔卷积码（RCPC）

从码率 $R=1/n$ 的母卷积码出发生成一系列码率不同的凿孔卷积码，使这些卷积码在码率上可以兼容，也就是说使高码率的凿孔卷积码的所有输出编码比特（未被凿孔的比特）都包含在低码率凿孔卷积码输出中，因此高码率凿孔卷积码嵌入在低码率凿孔卷积码中。

这在实际应用中非常有意义。因为高码率的凿孔卷积码被凿掉的比特较多，其性能退化也较严重，如果这时传输性能达不到所要求水平，则可以补充传输那些被凿孔掉的比特，使码率适当降低，从而达到的所要求的性能水平。这种技术在自适应通信，多码率传输和不均匀保护方面有重要应用价值。