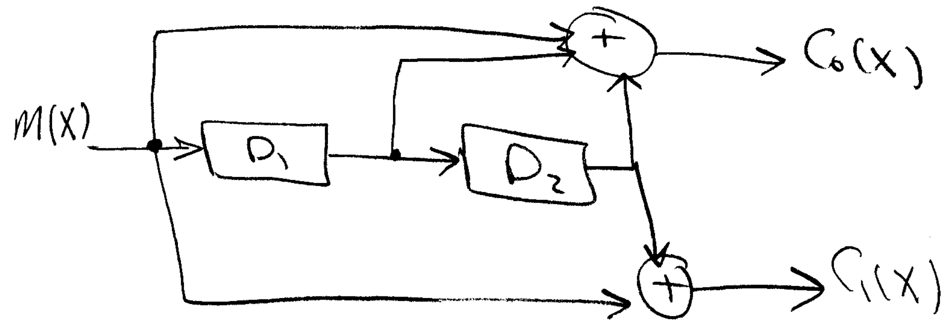
Ch. 6

Convolutional Codes

Example Rate $1/2$, $M=2$ Convolutional Encoder



In the above example,

- The memory depth of the registers is $M=2$
- For each one bit input, there are two bits output. $\Rightarrow$ Rate $R = k/n = 1/2$.
- Since the effect of any one data input lasts over $\nu = M+1 = 2+1 = 3$ bits
  $\Longrightarrow$ Constraint length $= \nu = M+1$
- The encoder above is a finite impulse response [FIR] encoder.
- The generator polynomials are

$$g_0(x) = 1 + x + x^2$$
$$g_1(x) = 1 + x^2$$

$$\Longrightarrow \quad C_0(x) = m(x)\, g_0(x), \quad C_1(x) = m(x)\, g_1(x)$$

- In general, the Memory depth M of a binary convolutional Code is:

$$M = \max \deg [g_0(x), \cdots, g_{n-1}(x)]$$

where $g_i(x)$ is the generator polynomial for $C_i(x)$.

6.2 Structural Properties of Convolutional Codes

- State Diagram and Trellis Representations.

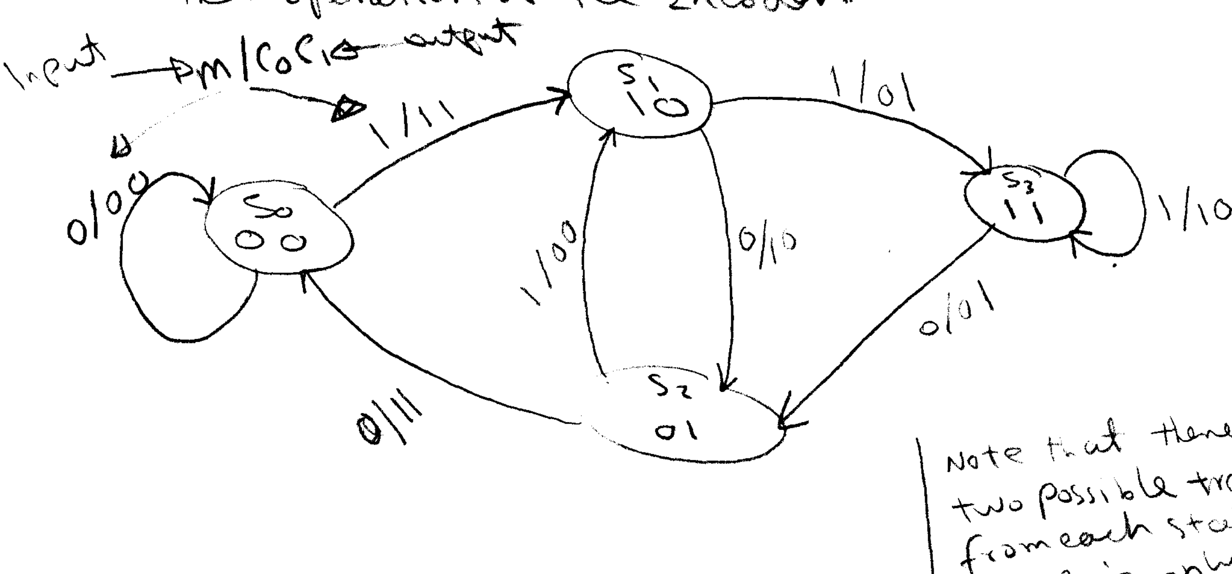
- State Diagram

There are 2^M states in an encoder with M memory elements.

- For the previous example, $2^2 = 4$ states
- Let us name the states

$$\left.\begin{aligned} S_0 &= 00 \\ S_1 &= 10 \\ S_2 &= 01 \\ S_3 &= 11 \end{aligned}\right\} \text{content of the shift Registers.}$$

- We can draw the state diagram from observing The operation of the Encoder.



Note that there are two possible transitions from each state since there is only one i... In general, 2^k branches will be going out from each state

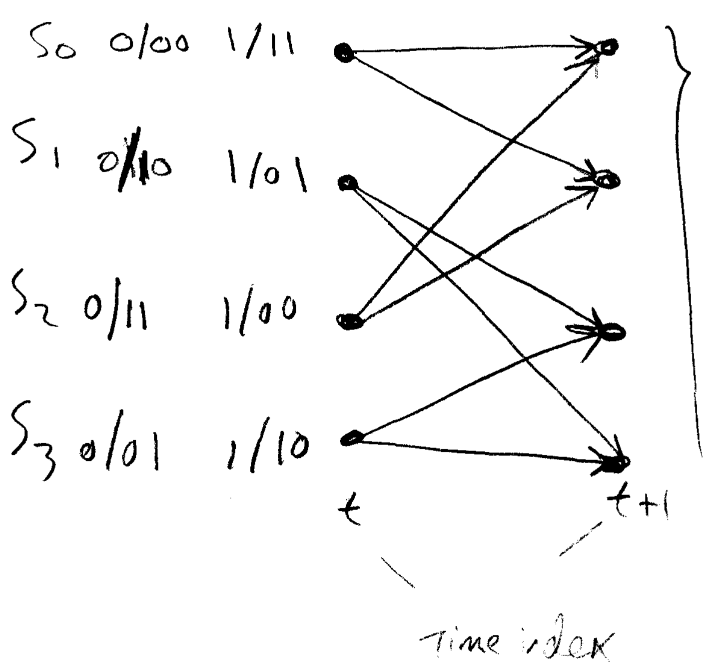
So, we can follow the state transition and know the output codeword for an input sequence.

For example, let $m = [0\,0\,1\,0\,1\,1\,0\,1]$ ← First In.

The output codeword will be (starting from state zero)

$C = [11\ 10\ 00\ 01\ 01\ 00\ 10\ 11]$ ← First out

Trellis Diagram



4-State Trellis Diagram. 2^k transition branch from each Node [state]

* The trellis diagram is very important in analysing the Hamming distance of the code.
* Also, Decoding is based on the Viterbi algorithm which is based on the trellis diagram.

* Convolutional Codes are Linear codes

* The Hamming distance properties of any two code sequences in the trellis are equivalent to the Hamming distance properties between some code sequence and the all-zeros code sequence.

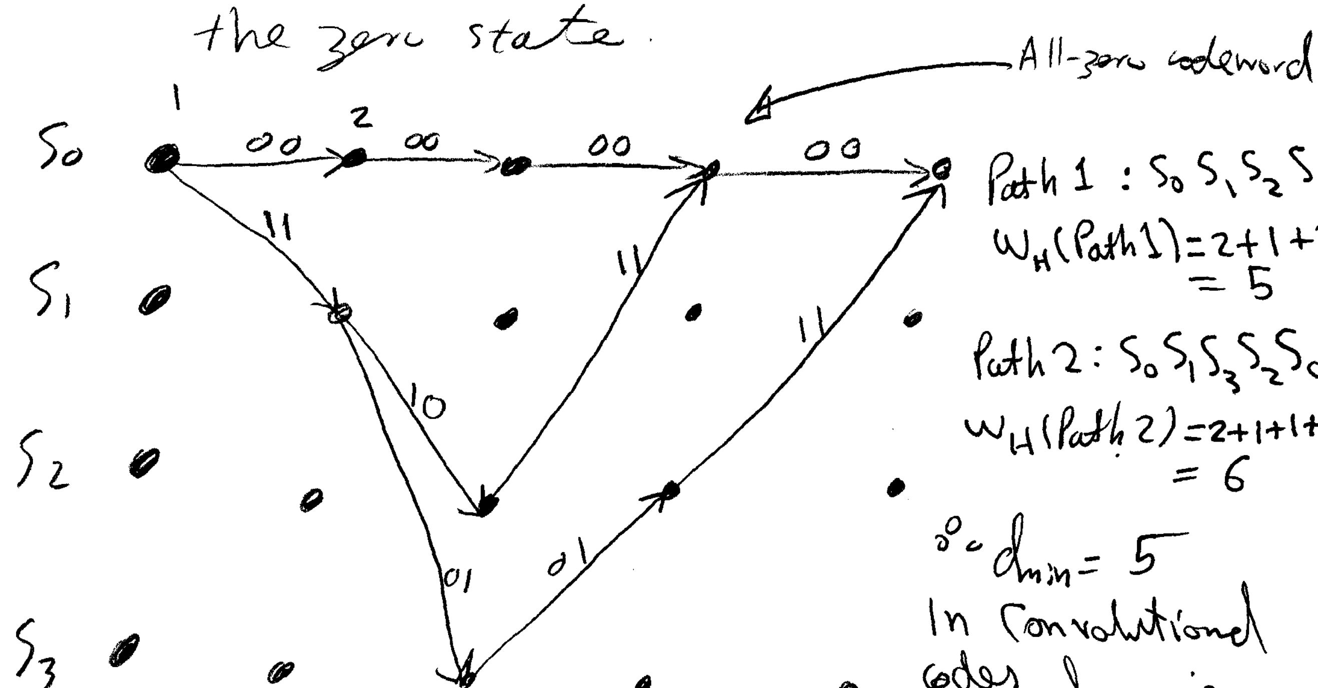
6.2 Structural Properties of Convolutional Codes. {Continue}

6.2.2 Transfer Functions

In order to find the distance properties of Convolution Codes, we need to drive the tranfer Function from the state diagram.

Method 1 Find the Hamming distance From the Trellis Diagram.

In order to find the minimum Hamming distance (d_{min}), we compare all possible codewords with all zero codeword. We can do this by diverging from all zero codeword and trying to return back to the zero state.



Path 1 : $S_0 S_1 S_2 S_0$

$W_H(\text{Path}1) = 2+1+2 = 5$

Path 2 : $S_0 S_1 S_3 S_2 S_0$

$W_H(\text{Path}2) = 2+1+1+2 = 6$

$\therefore d_{min} = 5$

In Convolutional codes d_{min} is called the free Hamming distance

$\boxed{(d_{free}) = d_f}$

Method 2 : Tranfer Function.

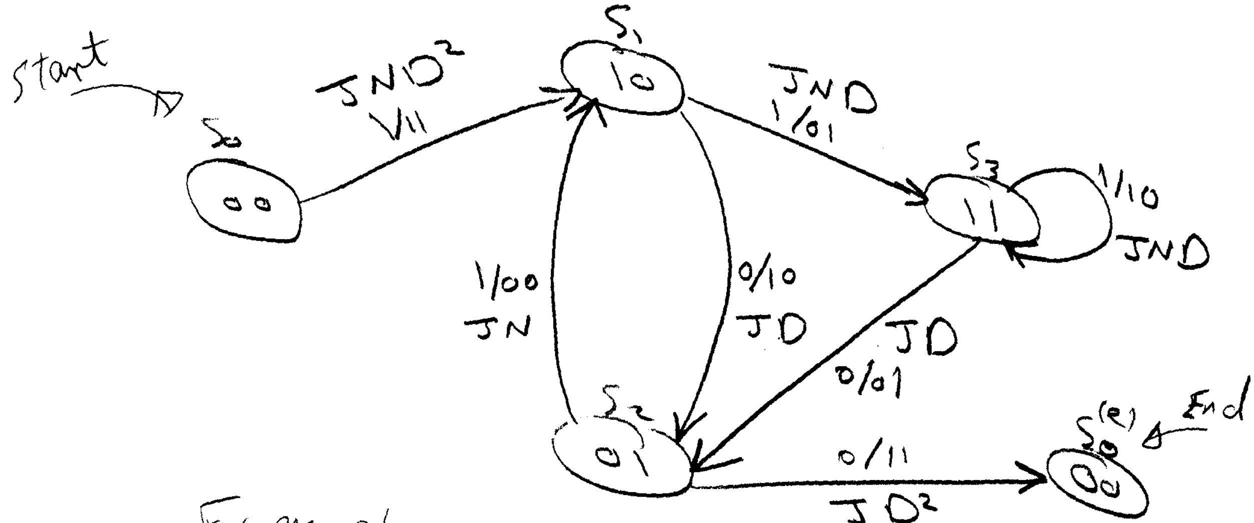
We introduce the following operators.

D : codeword weight operator.

N : Source symbol weight operator.

J : time index operator

The state diagram will be



For example, the path $S_0 S_1 S_3 S_3$ will have tranfer function $(JND^2)(JND)(JND)$

$= J^3 N^3 D^4$

$\Rightarrow$ wight of codeword = 4
wight of source symbol = 3
Path length = 3

The tranfer function from the start (S_0) to the end $(S_0^{(e)})$ Describes all possible Paths.

Method 1 $\rightarrow$ $T(J,N,D) = D^5 J^3 N + D^6 J^4 N^2 + D^6 J^5 N^2 + D^7 J^5 N^3 + \ldots$

Shortest Path of weight Five and length Three.

— Method 2: Find the transfer function by solving linear equations.

at $S_1 \longrightarrow X_1 = JND^2 X_0 + JN X_2$

at $S_2 \longrightarrow X_2 = JD X_1 + JD X_3$

at $S_3 \longrightarrow X_3 = JND X_1 + JND X_3$

at $S_0^{(e)} \longrightarrow X_0^{(e)} = JD^2 X_2$

Let $X_0 = 1$ and solve for $X_0^{(e)}$.

The answer will be

$$T(J,N,D) = \frac{D^5 N J^3}{1 - DNJ(1+J)}$$

To see the individual paths, apply long division to $T(J,N,D)$, we get

$$T(J,N,D) = D^5 J^3 N + D^6 J^4 (1+J) N^2 + \cdots + D^{j+5} J^{j+3} (1+J)^j N^{j+1} + \cdots$$

* The ~~Transfer~~ function supplies us with all the information we need to completely characterize the structure and performance of the code.

For example, the term

$$D^6 J^4 (1+J) N^2 = D^6 J^4 N^2 + D^6 J^5 N^2$$

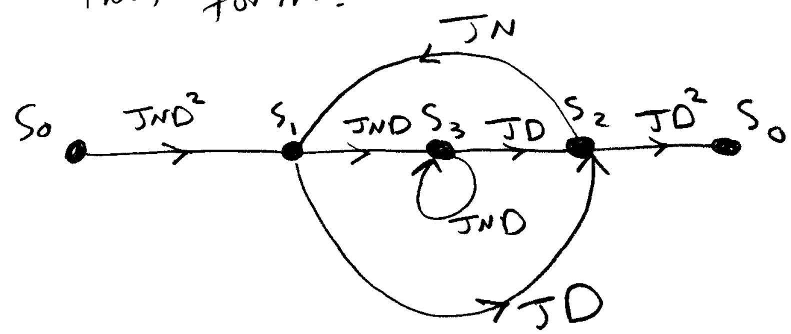
This term tells us that there are two paths of Hamming wieght 6 and both involve source sequence with Hamming wieght 2. One of them takes four steps through the trellis while the other takes five steps.

Transfer function of the convolutional codes

Method 3 Mason's Rule

we can also use Mason's Rule from signal flow graph Theory to find the transfer function of convolutional codes.

First, Redraw the state diagram in this form:



Manson's Rule

The transfer function is $T = \sum_{k=1}^{N} \frac{P_k \Delta_k}{\Delta}$

where,

N Number of paths from input to output

P_k Gain of the K_{th} path from input to output

Δ Determinant of the graph

Δ_k Co factor of the k_{th} path

where, $\Delta = 1 - (\sum$ all different loop gains$)$

$+ (\sum$ gain product of all combinations of 2 non-touching loops$)$

$- (\sum$ gain products of all combinations of 3 non-touching loops$)$

$+ \ldots\ldots$

and Δ_k = value of Δ for the signal flow graph not touching the kth forward path.

Analysing the state diagram of the encoder we find two forward paths; $P_1 = J^4 N^2 D^6$

$P_2 = J^3 N D^5$

~~Three~~ loops; $L_1 = JND$

$L_2 = J^3 N^2 D^2$ non-touching.

$L_3 = J^2 N D$

$$\therefore T = \frac{P_1 \Delta_1 + P_2 \Delta_2}{\Delta}$$

$$= \frac{J^4 N^2 D^6 \{1 - 0\} + J^3 D^5 N \{1 - JND\}}{1 - [JND + J^2 ND + J^3 N^2 D^2] + J^3 N^2 D^2}$$

$$= \frac{J^4 N^2 D^6 + J^3 D^5 N - J^4 N^2 D^6}{1 - [JND + J^2 ND]}$$

$$\boxed{T = \frac{J^3 D^5 N}{1 - DNJ[1 + J]}}$$

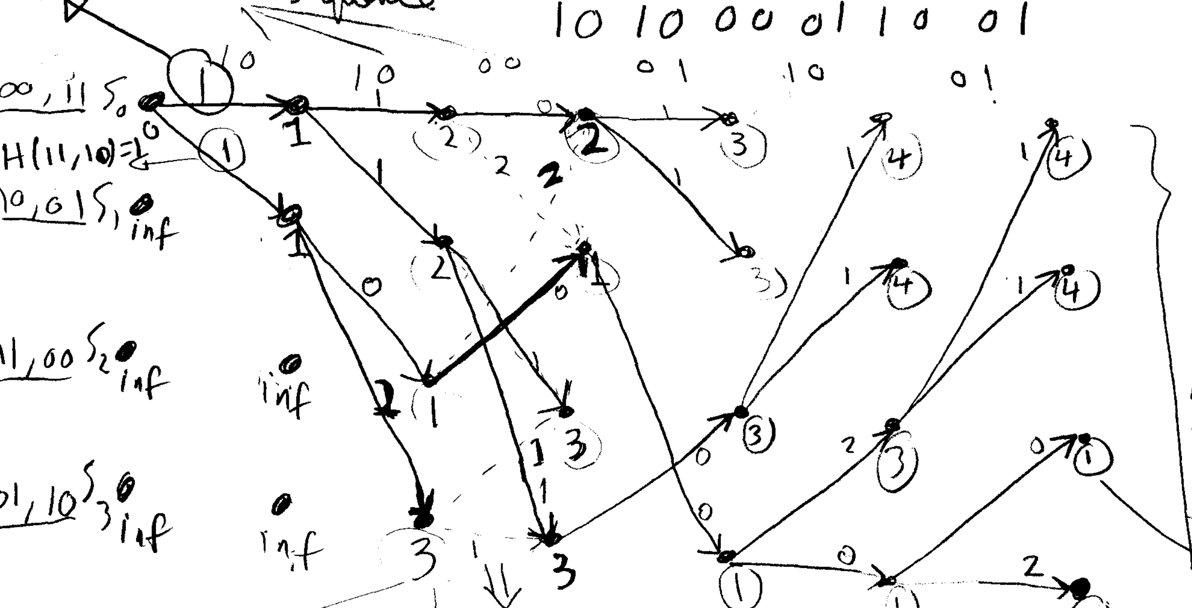
6.3 The Viterbi Algorithm.

- The viterbi algorithm is used to decode convolutional codes and any structure or system that can be described by a trellis
- It is a maximum likelihood decoding algorithm that selects the most probable path that maximizes the likelihood function.
- The algorithm is based on add-compare-select the best path each time at each state.

Example 6.3.2 For the convoultional code example in the previous lectures, starting from state zero, Decode the following received sequence

10 10 00 01 10 01

Hamming wight $H(00,10)=1$ Received sequence



add the wights of the path at each state

compare the two possible paths at each state and select the one with less comulative Hamming wight $\Rightarrow$ This is called the survived path

At the end of the trellis, select the path with the minimum comulative Hamming weight.

This is the survived path in this example

$\Rightarrow$ Decoded sequence is

$m = [1\,0\,1\,1\,1\,0]$.

- Trace back the path to find the information bits