# CT216 Convolution Code Analysis

G 34

April 2025

## Contents



# 1 What is Transfer Function and How to Find It

## 1.1 Analogy

Like an LTI system where we have some input which can be represented as a impulse train, and find a impulse response to that LTI system. Then we can convolve input with impulse response to get output. In time domain convolution is a multiplication in frequency domain by a transfer function.

In time Domain

$$y(t) = x(t) * h(t)$$

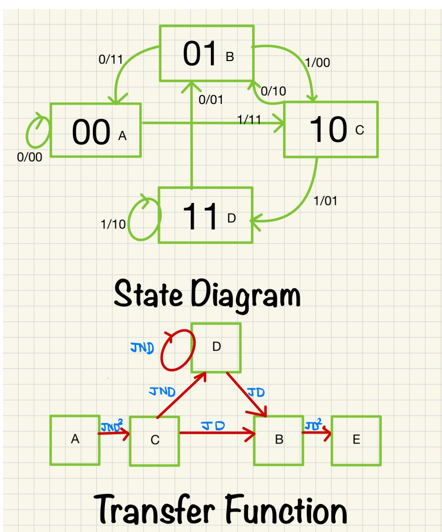
In Frequency Domain

$$Y(w) = X(w)H(w)$$

Similarly in Convolutional Codes time domain impulse response can be found by generator matrix passing a $[1, 0, 0, 0, ..., 0]$ as impulse and then recording it's output.This is it's impulse response. Then we can find it's output by just delay and linear combination of impulse response.
We can see it from a different perspective. Just like time and frequency we can have time and hamming distance as 2 sides of the same encoder.

## 1.2 Method of finding a Transfer Function for rate 1/2 $K_c = 3$

### 1.2.1 State Diagram




Here Exponent of J is a counter for path length.
Also Exponent of N represent's number of $1^s$ in the codeword.
And Exponent of D represents hamming distance for a codeword.

### 1.2.2 State Function

$$X_c = JND^2X_a$$
$$X_d = JND(X_c + X_d)$$
$$X_b = JD(X_c + X_d)$$
$$X_e = JD^2X_b$$

### 1.2.3 Transfer Function

$$\frac{X_e}{X_a} = \frac{J^3ND^5}{1 - JND}$$
$$T(J, N, D) = \sum_{i=0}^{\infty} J^{i+3}N^{i+1}D^{i+5}$$
$$d_{free} = 5$$
$$T(J, N, D) = \sum_{d=d_{free}}^{\infty} J^{d-2}N^{d-4}D^d$$

# 2 Derivation of Soft Decoding Parameters

## 2.1 Branch Metric

$$\mu_j^i = \sum_{m=1}^{n} r_{jm}(2C_{jm}^i - 1)$$
$$r_{jm} = \sqrt{e_c}(2C_{jm} - 1) + n_{jm}$$
$$n_{jm} -> N(0, \sigma^2)$$

Thus $r_{jm}$ follows the following distribution

$$r_{jm} -> N(\sqrt{e_c}(2C_{jm} - 1), \sigma^2)$$

Here, $\sigma^2 = \frac{N_o}{2}$

## 2.2 Correlation Metric

$$CM^i = \sum_{j=1}^{N} \mu_j^i$$

## 2.3 Finding Probability of First Error at distance d

Since Convolution Codes are LTI system. Thus all zeros codeword is a valid codeword. We will use this property in all the analysis.

$$P^d_{firstError} = P(CM^1 > CM^0)$$

Since we have transmitted an all zero codeword so its correlation metric would be lower compared to path with 'd' hamming distance from the sent codeword.

$$CM^1 > CM^0$$

$$P(CM^1 - CM^0 > 0) = P(\sum_{j=1}^{N}(\mu_j^1 - \mu_j^0))$$

$$P(CM^1 - CM^0 > 0) = P(2\sum_{j=1}^{N}\sum_{m=1}^{n} r_{jm}(C^1_{jm} - C^0_{jm}) > 0)$$

Since we have sequences differing at 'd' position's thus all other terms in the sum will evaluate to 0.

$$P(CM^1 - CM^0 > 0) = P(\sum_{l=1}^{d} r_l > 0)$$

Also $r_l$ is a normal random variable. Thus their sum is also a normal random variable.

$$\sum_{l=1}^{d} r_l -> N(\sqrt{e_c}\sum_{l=1}^{d}(2C_l - 1), d\sigma^2)$$

Since input was all zero codeword, thus $C_l = 0$.

$$X = \sum_{l=1}^{d} r_l -> N(-d\sqrt{e_c}, d\sigma^2)$$

$$Y = \frac{X + d\sqrt{e_c}}{\sigma\sqrt{d}}$$

For $X > 0$ it implies that $Y > \sqrt{\frac{de_c}{\sigma^2}}$. Thus $P(X > 0) = P(Y > \sqrt{\frac{2e_c d}{N_0}}) = Q(\sqrt{\frac{2e_c d}{N_0}})$

$$P^d_{firstError} = Q(\sqrt{\frac{2e_c d}{N_0}})$$

For a approximation we can write Q function with upper bound as follows:

$$P^d_{firstError} = Q(\sqrt{\frac{2e_c d}{N_0}}) < e^{-\frac{e_c d}{N_0}}$$

### 2.4 Finding Probability of First error irrespective of length

$$P_{firstError} < \sum_{d=d_{free}}^{\infty} (number\, of\, possible\, path\, with\, hamming\, distance\, d) * P_{firstError}^{d}$$

This is just considering all possible paths of distance 'd' where d is from $d = d_{free}$ till $d = \infty$.

$$P_{firstError} < \sum_{d=d_{free}}^{\infty} a_d * P_{firstError}^{d}$$

Here $a_d$ is the number of paths with hamming distance 'd' from actual codeword. We can write Probablity in another way as follows.

$$P_{firstError} < \sum_{d=d_{free}}^{\infty} a_d (e^{-\frac{e_c}{n_0}})^d$$

Here we can write $e^{-\frac{e_c}{n_0}}$ as $D$ just as in transfer function. Also since $a_d$ can be found from transfer function we can write Probablity in terms of transfer function.

$$P_{firstError} < \sum_{d=d_{free}}^{\infty} a_d (D)^d = T(J, N, D) | J = 1, N = 1, D = e^{-\frac{e_c}{N_0}}$$

### 2.5 Bit Error Rate

$$BER^d = a_d (Number\, of\, mismatch\, bits) P_{firstError}^{d}$$

Thus over all path length's it will be the following:

$$BER < \sum_{d=d_{free}}^{\infty} a_d (Number\, of\, mismatch\, bits) P_{firstError}^{d}$$

We know that in transfer function exponent of N represent's number of 1's in the sequence. Since we are transmitting all zero codeword which is also the input. Thus there will be exponent of N number of mismatch at any distance 'd'. Thus if we want to include transfer function then we can write it as follows:

$$BER = \sum_{d=d_{free}}^{\infty} a_d g(d) D^d$$

This form can be acheived by keeping $D = e^{-\frac{e_c}{N_0}}$ and $J = 1, N = 1$

$$BER < \frac{d(T(J, N, D))}{dN} | J = 1, N = 1, D = e^{-\frac{e_c}{N_0}}$$

### 2.5.1 BER for our example

In our example Transfer Function is as follows:

$$T(J,N,D) = \sum_{d=d_{free}}^{\infty} J^{d-2}N^{d-4}D^{d}$$

Thus BER is calculated as follows:

$$BER < \frac{d(T(J,N,D))}{dN}|J=1, N=1, D=e^{-\frac{e_c}{N_0}}$$

$$\frac{dT}{dN} = \sum_{d=5}^{\infty} J^{d-2}(d-4)N^{d-5}D^{d}$$

$$BER < \sum_{d=5}^{\infty}(d-4)D^{d}$$

This is a arithmetico-Geometric Progression. Thus,

$$BER = \frac{D^5}{(1-D)^2}$$

Here $D = e^{-\frac{e_c}{N_0}} = e^{-rate\frac{e_b}{N_0}}$

$$BER = \frac{(e^{-rate\frac{e_b}{N_0}})^5}{(1-e^{-rate\frac{e_b}{N_0}})^2}$$

# 3 Derivation of Hard Decoding Parameters

## 3.1 Probability of first error event for length d

Since in hard decision decoding there is hamming distance as a comparison between 2 codeword sequences.
If a sequence with hamming distance d is received with error's less than $\frac{d+1}{2}$ then it is solvable. Else not solvable.

$$P^{d}_{firstError} = \sum_{k=\frac{d+1}{2}}^{d} \binom{d}{k} (P_{bitFlip})^{k}(1-P_{bitFlip})^{d-k}$$

This can be approximated by the below formula:

$$P^{d}_{firstError} = \sqrt{4P_{bitFlip}(1-P_{bitFlip})}$$

$$P_{bitFlip} = Q(\sqrt{\frac{2e_c}{N_0}})$$

### 3.2 Probability of first error irrespective of d

$$P_{firstError} < \sum_{d=d_{free}}^{\infty} a_d P_{firstError}^d$$

$$P_{firstError} < \sum_{d=d_{free}}^{\infty} a_d(\sqrt{4P_{bitFlip}(1-P_{bitFlip})})^d$$

Here we can write $D = \sqrt{4P_{bitFlip}(1-P_{bitFlip})}$

$$P_{firstError} < T(J,N,D)|J=1, N=1, D=\sqrt{4P_{bitFlip}(1-P_{bitFlip})}$$

### 3.3 Bit Error Rate

$$BER^d = a_d(Number\,of\,mismatch\,bits)P_{firstError}^d$$

We know that in transfer function exponent of N represent's number of 1's in the sequence. Since we are transmitting all zero codeword which is also the input. Thus there will be exponent of N number of mismatch at any distance 'd'. Thus if we want to include transfer function then we can write it as follows:

$$BER < \frac{d(T(J,N,D))}{dN}|J=1, N=1, D=\sqrt{4P_{bitFlip}(1-P_{bitFlip})}$$

#### 3.3.1 BER for our example

In our example Transfer Function is as follows:

$$T(J,N,D) = \sum_{d=d_{free}}^{\infty} J^{d-2}N^{d-4}D^d$$

Thus BER is calculated as follows:

$$BER < \frac{d(T(J,N,D))}{dN}|J=1, N=1, D=\sqrt{4P_{bitFlip}(1-P_{bitFlip})}$$

$$\frac{dT}{dN} = \sum_{d=5}^{\infty} J^{d-2}(d-4)N^{d-5}D^d$$

$$BER < \sum_{d=5}^{\infty}(d-4)D^d$$

This is a arithmetico-Geometric Progression. Thus,

$$BER = \frac{D^5}{(1-D)^2}$$

Here $D = \sqrt{4P_{bitFlip}(1 - P_{bitFlip})}$
Also Probablity of bit flip is:

$$P_{bitFlip} = Q(\sqrt{\frac{2e_c}{N_0}})$$