CT216

introduction to communication system

Lab group 2

- We declare that
  - The work that we are presenting is our own work.
  - We have not copied the work (the code, the results, etc.) that someone else has done.
  - Concepts, understanding and insights we will be describing are our own.
  - We make this pledge truthfully. We know that violation of this solemn pledge can
  - carry grave consequences.

1. Divyarajsinh Chundavat – 202201155 Divyaraj
2. Meet Katharotiya -202201157 Meet.
3. Kishan Patel – 202201159 K.K.Patel
4. Smit Godhani-202201162 Smit
5. Jay Goyani-202201163 Jay
6. Krisha Brahmbhatt – 202201164
7. Het Gandhi – 202201167 Het
8. Jyot vasava – 202201169
9. Parshwa Modi – 202201165
10. Parth Vadodaria – 202201174 Parth
11. Harshvardhan Vajani – 202201413 HVajani

Soft Decision Decoding

Analysis

Appendix

- Convolution Encoder is a finite state machine, And the optimum decoder is a maximum likelihood sequence estimator (MLSE), which helps us to find the most likely transmitted sequence for the received sequence.
- If the Soft-Decision Decoding is employed and the code sequence is transmitted by BPSK modulation then the input to the decoder is:

$$r_{jm} = \sqrt{\varepsilon_c}\,(2c_{jm} - 1) + n_{jm} \qquad \text{……… (8-2-9)}$$

❑Here ,

c_{jm} : coded symbol related to m^{th} symbol of j^{th}branch and it's components are c_{jm1}, c_{jm2} etc.

r_{jm} : output from the demodulator i.e. the Viterbi decoder inputs corresponding to m^{th} symbol of j^{th} branch in trellis.

n_{jm}: represents the additive white gaussian noise that affects the reception of m^{th} symbol of j^{th} branch in trellis.

- <u>Branch Metric :</u>

A branch metric for the j^{th} branch of the i^{th} path through the trellis is defined as the logarithm of the joint probability of the sequence conditioned on transmitted sequence

$$\mu_j^{(i)} = \log P\,(\{r_{jm}\}|\{c_{jm}\}) \qquad \text{……… (8-2-10)}$$

Appendix

- Now, $P\left(\{r_{jm}\}|\{c_{jm}{}^{i}\}\right) = P\left(r_{jm1}r_{jm2}r_{jm3}\ldots|\{c_{jm1}{}^{i}c_{jm2}{}^{i}c_{jm3}{}^{i}\ldots\}\right)$
- Since $r_{ji}'s$ are independent of each r_{jm} for any i $\neq m$,

$$P\left(\{r_{jm}\}|\{c_{jm}{}^{i}\}\right) = \Pi_{m=1}^{m=n} P\left(r_{jm}|\{c_{jm}{}^{i}\}\right) \qquad \text{......... (a)}$$

- Considering that the reception of r_{ji} is only being affected by c_{ji}

$$P\left(r_{jm}|\{c_{jm}{}^{i}\}\right) = P\left(r_{jm}|c_{jm}{}^{i}\right)$$

- Now, demodulator output is described statistically by the pdf

$$P\left(r_{jm}|c_{jm}{}^{i}\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}\left[n_{jm}\right]^2\right) \qquad \text{......... (b)}$$

- Therefore,

$$P\left(r_{jm}|c_{jm}{}^{i}\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}\left[r_{jm} - \sqrt{\varepsilon_c}\left(2c_{jm} - 1\right)\right]^2\right) \qquad \text{......... (8-2-13)}$$

- Where, $\sigma^2 = \frac{N_0}{2}$ is variance of Additive White Gaussian Noise.
- Using equation (a), (b) and (8-2-13) into equation (8-2-10) and by neglecting the common terms to all branch metrics then branch metric for the j^{th} branch of the i^{th} path may be expressed as

$$\mu_J{}^{(i)} = \sum_{m=1}^{m=n} r_{jm}\left(2c_{jm}{}^{i} - 1\right) \qquad \text{......... (8-2-14)}$$

Appendix

- <u>Path Metric :</u>

  A path metric for the i^{th} path containing B branches through the trellis is defined as

$$PM^{(i)} = \sum_{j=1}^{j=B} \mu_j^{(i)} \qquad \text{......... (c)}$$

- The guidelines for choosing between two paths within trellis is to opt for the one with higher metric. This enhances the likelihood of making accurate decision or conversely if diminishes the probability of errors in the sequence of information bits.
- It is also called convolution metric of the path as well.

$$CM^{(i)} = \sum_{j=1}^{j=B} \sum_{m=1}^{m=n} r_{jm}\left(2c_{jm}{}^{i} - 1\right) \qquad \text{......... (8-2-16)}$$

- This metric is nothing but essentially the summation of all branch metrics of the i^{th} path.

- <u>Probability of error for Soft-Decision-Decoding :</u>
- To derive the probability of error for convolutional codes, linearity property can be used to simplify the derivation.
- That is, we can assume that all-zero sequence is transmitted and we determine the probability of error in favor of other sequence.

❑Now, for all-zero path

$$CM^{(0)} = \sum_{j=1}^{j=B} \sum_{m=1}^{m=n} (-\sqrt{\varepsilon_c} + n_{jm})(-1)$$

Appendix

$$CM^{(0)} = \sqrt{\varepsilon_c}Bn - \sum_{j=1}^{j=B}\sum_{m=1}^{m=n}(n_{jm}) \quad \text{......... (8-2-17)}$$

- Now , It the incorrect path that merges with all-zero path differs from 'd' bits will be having it's correlation metric $CM^{(1)}$, then the probability of error in pairwise comparison of $CM^{(0)}$ and $CM^{(1)}$ is,

$$P_2(d) = P\left(CM^{(1)} \geq CM^{(0)}\right) = P\left(CM^{(1)} - CM^{(0)} \geq 0\right)$$

- Using eq(8-2-16)

$$P_2(d) = P\left(2\sum_{j=1}^{j=B}\sum_{m=1}^{m=n} r_{jm}\left(c_{jm}{}^{(1)} - c_{jm}{}^{(0)}\right) \geq 0\right) \quad \text{......... (8-2-18)}$$

- Since the coded bits in the two paths we identical except in the ' d ' positions , this equation can be written as

$$P_2(d) = P\left(\sum_{l=1}^{l=d} r_l' \geq 0\right) \quad \text{......... (8-2-19)}$$

❑Here, r_l' represents input to the decoder for d^{th} bits.

- From equation (1)

$$r_{jm} = \sqrt{\varepsilon_c}\ (2c_{jm} - 1) + n_{jm}$$

Where, $n_{jm} \sim N\left(0, \frac{N_0}{2}\right) \rightarrow r_{jm}/r_l' \sim N\left(-\sqrt{\varepsilon_c}, \frac{N_0}{2}\right)$

❑Now, from Central Limit Theorem :

$$\sum_{l=1}^{l=d} r_l' \sim N\left(-d\sqrt{\varepsilon_c}, \frac{dN_0}{2}\right)$$

Appendix

- Therefore, $P_2(D) = \mathrm{P}\left[\frac{\sum_{l=1}^{d} r_l' + \mathrm{d}\sqrt{\varepsilon_c}}{\sqrt{\frac{dN_0}{2}}} \geq \frac{\mathrm{d}\sqrt{\varepsilon_c}}{\sqrt{\frac{dN_0}{2}}}\right] = Q(\sqrt{\frac{2d\varepsilon_c}{N_0}})$ .... (d)

$= Q(\sqrt{2\gamma_b R_c d})$ .... (8-2-20)

❑Where, $\gamma_b = \frac{\varepsilon_b}{N_0}$ is received SNR per bit

R_c is code rate

- Now , there are many paths with different distances that merge with the all-zero path at given node. Transfer function provides (T(D)) a complex description of all such paths.
- Thus, we can sum the error probability (eq(10)) for all such possible paths which will correct into the first event error probability in the form.

$$P_e \leq \sum_{d=d_{free}}^{INF} a_d P_2(d)$$

$$\leq \sum_{d=d_{free}}^{INF} a_d \, Q(\sqrt{2\gamma_b R_c d})$$ .... (8-2-21)

❑Where,

d_{free} - is smallest Hamming distance between any two particular codeword

a_d - Number of paths of distance d (which particularly is the number of 1's in that path)

Appendix

❑Now, applying chernoff's inequality we get $P(x \geq a) \leq \exp(-at)E[e^{tX}]$, which further yields to the result

$$Q(x) \leq \exp(\frac{-x^2}{2}) \quad \text{......... (e)}$$

$$\rightarrow Q\left(\sqrt{2\gamma_b R_c d}\right) \leq \exp(-\gamma_b R_c d) = D^d|_{D=e^{-\gamma_b R_c}} \quad \text{......... (8-2-22)}$$

- From equations (11) and (12)

$$P_e < \sum_{d=d_{free}}^{INF} a_d D^d \quad \text{......... (f)}$$

- Now, bit error probability can be more useful measure of performance. This probability can be upper bounded by the procedure used in bounding the first event error probability if we multiply $p2\ (d)$ by # of incorrectly decoded information bits.
- The average bit error probability is upper bounded by multiplying each $p2\ (d)$ by the corresponding # of incorrectly decoded bits for each possible incorrect path.

❑Now, $T(D, N) = \sum_{d=d_{free}}^{INF} a_d\ D^d\ N^{f(d)}$[8-2-24]. Here, exponent of N represents the number of 1's in the path with d distance.

- Therefore,

$$\frac{T(D,N)}{dN} = \sum_{d=d_{free}}^{INF} a_d \mathrm{f(d)} D^d$$

$$\frac{T(D,N)}{dN} = \sum_{d=d_{free}}^{INF} \beta_d D^d \quad \text{......... (8-2-25)}$$

Appendix

- Thus, the bit error probability for k=1 is

$$P_b < \sum_{d=d_{free}}^{INF} \beta_d P_2(d)$$

$$P_b < \sum_{d=d_{free}}^{INF} \beta_d Q\left(\sqrt{2\gamma_b R_c d}\right) \qquad \text{......... (8-2-26)}$$

Hard Decision Decoding

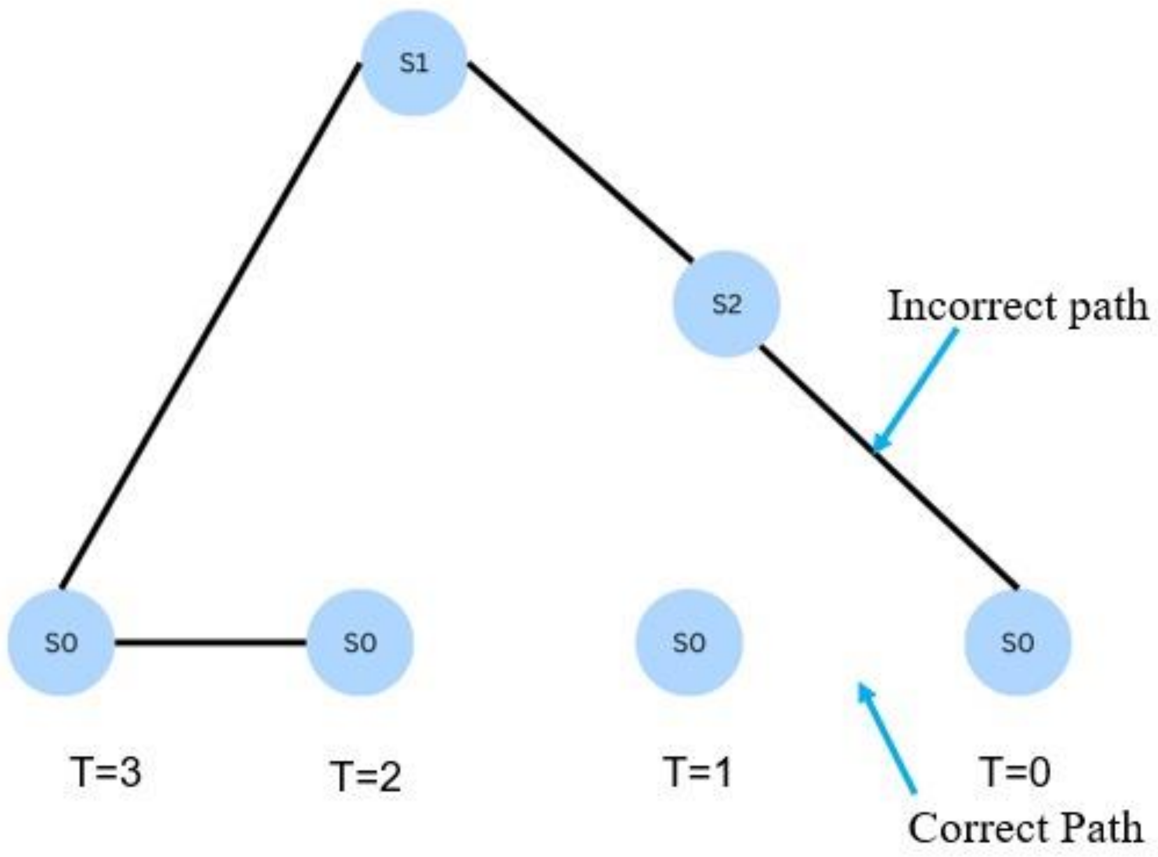
Analysis

Appendix

- We will analyze the performance of maximum likelihood decoding for a convolutional code over binary symmetric channel.
- Without loss of generality, we assume that the all zero codeword 0 is transmitted .
- A first event error happens at an arbitrary time t if the all zero path is eliminated for the first time in favor of an incorrect path.
- Assuming that the incorrect path has weight d, a first event error happens with probability

$$P_2(\mathrm{D}) = \begin{cases} \sum_{k=(d+1)/2}^{d} \binom{d}{k} p^k (1-p)^{d-k} & \text{If d is odd} \quad \text{......... (8-2-28)} \\ \sum_{k=(d+1)/2}^{d} \binom{d}{k} p^k (1-p)^{d-k} + \frac{1}{2}\binom{d}{k} p^k (1-p)^{d-k} & \text{If d is even} \quad \text{......... (8-2-29)} \end{cases}$$

Appendix



- All incorrect paths of length t branches or less can cause a first event error at time t.
- Thus the first event error probability at time t can be bounded using union bound by the sum of the error probabilities of each of these paths.

- If all incorrect paths of length greater than t are also included, then the first event error probability at any time t can be bounded by

$$P_e < \sum_{d=d_{free}}^{d} a_d \, P_2(D) \qquad \text{......... (8-2-30)}$$

- Where a_d is the number of codewords of weight d.

Appendix

- Instead of using the expressions for P2(d) given in (8-2-28) and (8-2-29), we can use the upper bound,

$$P_2(\mathrm{D}) < [4p\ (1-p)\ ]^{d/2}$$ .... (8-2-31)

- Use of this bound in (8-2-30) yields a looser upper bound on the first-event error probability, in the form,

$$P_e < \sum_{d=d_{free}}^{d} a_d\ [4p\ (1-p)\ ]^{d/2}$$
$$< \mathrm{T(D)}\big|_{D=\sqrt{4p(1-p)}}$$

.... (8-2-32)

Appendix

For odd d,

$$P_2(d) = \sum_{e=\frac{d+1}{2}}^{d} \binom{d}{e} p^e (1-p)^{d-e}$$

$$\leq \sum_{e=\frac{d+1}{2}}^{d} \binom{d}{e} p^{d/2} (1-p)^{d/2}$$

$$= p^{d/2}(1-p)^{d/2} \sum_{e=\frac{d+1}{2}}^{d} \binom{d}{e}$$

$$= p^{d/2}(1-p)^{d/2} \sum_{e=0}^{d} \binom{d}{e}$$

$$= 2^d p^{d/2}(1-p)^{d/2} \qquad \text{......... (a)}$$

For even d,

$$P_2(d) = \sum_{e=\frac{d}{2}+1}^{d} \binom{d}{e} p^e (1-p)^{d-e} + \frac{1}{2}\binom{d}{e} p^{d/2}(1-p)^{d/2}$$

$$< \sum_{e=\frac{d}{2}}^{d} \binom{d}{e} p^e (1-p)^{d-e}$$

$$< \sum_{e=\frac{d}{2}}^{d} \binom{d}{e} p^{d/2} (1-p)^{d/2}$$

$$= p^{d/2}(1-p)^{d/2} \sum_{e=0}^{d} \binom{d}{e} p^{d/2}$$

$$= 2^d p^{d/2}(1-p)^{d/2} \qquad \text{......... (b)}$$

Appendix

- The bit error probability can be bounded by

$$P_b < \sum_{d=d_{free}}^{\infty} \beta_d \, P_2(d) \qquad \text{......... (8-2-33)}$$

- where β_d is the total number of nonzero information bits on all weight-d paths, divided by the number of information bits k per unit time.
- The $\{\beta_d\}$ are the coefficients in the expansion of the derivative of T(D, N), evaluated at N = 1. For P_2(d), we may use either the expressions given in (8-2-28) and (8-2-29) or the upper bound in (8-2-31). If the latter is used, the upper bound on P_b, can be expressed as,

$$P_b < \frac{dT(D,N)}{dN}\bigg|_{N=1, D=\sqrt{4p(1-p)}} \qquad \text{......... (8-2-34)}$$

- When k >1, the results given in (8-2-33) and (8-2-34) for P_b, should be divided by k.

ThankYou !!!