An introduction to coding theory

Adrish Banerjee

Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, Uttar Pradesh
India

Feb. 13, 2017

Lecture #9B: Convolutional codes: Distance properties

Outline of the lecture

- Weight enumerating function calculation

Distance properties

- The minimum free distance of a convolutional code

$$d_{\text{free}} \stackrel{\Delta}{=} \min_{\mathbf{u},\mathbf{u}'} d(\mathbf{v},\mathbf{v}') : \mathbf{u} \neq \mathbf{u}'$$

where $\mathbf{v}$ and $\mathbf{v}'$ are the codewords corresponding to the information sequences $\mathbf{u}$ and $\mathbf{u}'$ respectively.

Distance properties

- The minimum free distance of a convolutional code

$$d_{\text{free}} \stackrel{\Delta}{=} \min_{\mathbf{u},\mathbf{u}'} d(\mathbf{v},\mathbf{v}') : \mathbf{u} \neq \mathbf{u}'$$

  where $\mathbf{v}$ and $\mathbf{v}'$ are the codewords corresponding to the information sequences $\mathbf{u}$ and $\mathbf{u}'$ respectively.
- d_{free} is the minimum Hamming distance between any two code sequences in the code.

Distance properties

- The minimum free distance of a convolutional code

$$d_{\text{free}} \stackrel{\Delta}{=} \min_{\mathbf{u},\mathbf{u}'} d(\mathbf{v},\mathbf{v}') : \mathbf{u} \neq \mathbf{u}'$$

  where $\mathbf{v}$ and $\mathbf{v}'$ are the codewords corresponding to the information sequences $\mathbf{u}$ and $\mathbf{u}'$ respectively.
- d_{free} is the minimum Hamming distance between any two code sequences in the code.
- d_{free} is also the minimum weight non-zero sequence, i.e.

$$d_{\text{free}} = \min\{w(\mathbf{v}) : \mathbf{u} \neq 0\}$$

Weight enumerating function calculation

- The performance of a convolutional code depends on its weight distribution.

Weight enumerating function calculation

- The performance of a convolutional code depends on its weight distribution.
- The weight distribution of a convolutional code is obtained by modifying the state diagram as follows:

Weight enumerating function calculation

- The performance of a convolutional code depends on its weight distribution.
- The weight distribution of a convolutional code is obtained by modifying the state diagram as follows:
  - All zero state is split into two states; initial state and final state and self loop around the all zero state is removed.

Weight enumerating function calculation

- The performance of a convolutional code depends on its weight distribution.
- The weight distribution of a convolutional code is obtained by modifying the state diagram as follows:
  - All zero state is split into two states; initial state and final state and self loop around the all zero state is removed.
  - Each branch is then labeled with a branch gain X^i, where i is the weight of the n output bits on that branch.

Weight enumerating function calculation

- The performance of a convolutional code depends on its weight distribution.
- The weight distribution of a convolutional code is obtained by modifying the state diagram as follows:
  - All zero state is split into two states; initial state and final state and self loop around the all zero state is removed.
  - Each branch is then labeled with a branch gain X^i, where i is the weight of the n output bits on that branch.
  - Each path connecting the initial state to the final state is a non-zero code sequence.

Weight enumerating function calculation

- The performance of a convolutional code depends on its weight distribution.
- The weight distribution of a convolutional code is obtained by modifying the state diagram as follows:
  - All zero state is split into two states; initial state and final state and self loop around the all zero state is removed.
  - Each branch is then labeled with a branch gain X^i, where i is the weight of the n output bits on that branch.
  - Each path connecting the initial state to the final state is a non-zero code sequence.
  - The path gain is the product of the branch gains along a path.

Weight enumerating function calculation

- The performance of a convolutional code depends on its weight distribution.
- The weight distribution of a convolutional code is obtained by modifying the state diagram as follows:
  - All zero state is split into two states; initial state and final state and self loop around the all zero state is removed.
  - Each branch is then labeled with a branch gain X^i, where i is the weight of the n output bits on that branch.
  - Each path connecting the initial state to the final state is a non-zero code sequence.
  - The path gain is the product of the branch gains along a path.
  - The weight of a code sequence is the power of X in the path gain of the corresponding path.

Weight enumerating function calculation

- The weight distribution function of a convolutional code is obtained by applying Mason's gain formula to compute the generating function

$$T(X) = \sum_i A_i X^i$$

of the modified state diagram, where A_i is the number of code sequences of weight i.

Weight enumerating function calculation

- The weight distribution function of a convolutional code is obtained by applying Mason's gain formula to compute the generating function

$$T(X) = \sum_i A_i X^i$$

of the modified state diagram, where A_i is the number of code sequences of weight i.

- *Forward path*: a path connecting the initial state to the final state that does not go through any state twice.

Weight enumerating function calculation

- The weight distribution function of a convolutional code is obtained by applying Mason's gain formula to compute the generating function

$$T(X) = \sum_i A_i X^i$$

of the modified state diagram, where A_i is the number of code sequences of weight i.

- *Forward path*: a path connecting the initial state to the final state that does not go through any state twice.

- *Loop*: a closed path that starts and ends in the same state without going over any other state twice.

Weight enumerating function calculation

- The weight distribution function of a convolutional code is obtained by applying Mason's gain formula to compute the generating function

$$T(X) = \sum_i A_i X^i$$

  of the modified state diagram, where A_i is the number of code sequences of weight i.
- *Forward path*: a path connecting the initial state to the final state that does not go through any state twice.
- *Loop*: a closed path that starts and ends in the same state without going over any other state twice.
- Two or more loops are non-touching if they don't have any states in common.

Weight enumerating function calculation

- Let F_i denote the gain of the i^{th} forward path.

Weight enumerating function calculation

- Let F_i denote the gain of the i^{th} forward path.
- Let C_i denote the gain of the i^{th} loop.

Weight enumerating function calculation

- Let F_i denote the gain of the i^{th} forward path.
- Let C_i denote the gain of the i^{th} loop.
- Let $\{i\}$ denote the set of all loops.

Weight enumerating function calculation

- Let F_i denote the gain of the i^{th} forward path.
- Let C_i denote the gain of the i^{th} loop.
- Let $\{i\}$ denote the set of all loops.
- Let $\{i,j\}$ denote the set of all pairs of non-touching loops.

Weight enumerating function calculation

- Let F_i denote the gain of the i^{th} forward path.
- Let C_i denote the gain of the i^{th} loop.
- Let $\{i\}$ denote the set of all loops.
- Let $\{i,j\}$ denote the set of all pairs of non-touching loops.
- Let $\{i,j,k\}$ denote the set of all triples of non-touching loops, and so on., then define

$$\Delta = 1 - \sum_{\{i\}} C_i + \sum_{\{i,j\}} C_i C_j - \sum_{\{i,j,k\}} C_i C_j C_k + \cdots$$

Weight enumerating function calculation

- Let F_i denote the gain of the i^{th} forward path.
- Let C_i denote the gain of the i^{th} loop.
- Let $\{i\}$ denote the set of all loops.
- Let $\{i,j\}$ denote the set of all pairs of non-touching loops.
- Let $\{i,j,k\}$ denote the set of all triples of non-touching loops, and so on., then define

$$\Delta = 1 - \sum_{\{i\}} C_i + \sum_{\{i,j\}} C_i C_j - \sum_{\{i,j,k\}} C_i C_j C_k + \cdots$$

- Let G_i be the graph obtained by removing all the states on the i^{th} forward path and all the branches connected to these states, and let Δ_i be defined similarly as Δ for the graph G_i.

Weight enumerating function calculation

- Mason's gain formula is given by

$$T(X) = \frac{\sum_i F_i \Delta_i}{\Delta}$$

Weight enumerating function calculation

- Mason's gain formula is given by

$$T(X) = \frac{\sum_i F_i \Delta_i}{\Delta}$$

- The modified state diagram can be augmented by labeling each branch corresponding to nonzero message bit with Y and labeling every branch with Z.

Weight enumerating function calculation

- Mason's gain formula is given by

$$T(X) = \frac{\sum_i F_i \Delta_i}{\Delta}$$

- The modified state diagram can be augmented by labeling each branch corresponding to nonzero message bit with Y and labeling every branch with Z.
- The augmented transfer function is given by

$$T(X, Y, Z) = \sum_{i,j,l} A_{i,j,l} X^i Y^j Z^l$$

where $A_{i,j,l}$ is the number of code sequences of weight i, whose corresponding information sequence has weight j, and which has length l branches.

Weight enumerating function calculation

- Input-output weight enumerating function (IOWEF): It is a property of the encoder.

Weight enumerating function calculation

- Input-output weight enumerating function (IOWEF): It is a property of the encoder.
- An alternative version of IOWEF that contains only information about input and output weights but not the length of each codeword is obtained as

$$T(X, Y) = T(X, Y, Z)|_{Z=1}$$

Weight enumerating function calculation

- Input-output weight enumerating function (IOWEF): It is a property of the encoder.
- An alternative version of IOWEF that contains only information about input and output weights but not the length of each codeword is obtained as

$$T(X, Y) = T(X, Y, Z)|_{Z=1}$$

- Weight enumerating function (WEF): It is a code property.

Weight enumerating function calculation

- Input-output weight enumerating function (IOWEF): It is a property of the encoder.
- An alternative version of IOWEF that contains only information about input and output weights but not the length of each codeword is obtained as

$$T(X, Y) = T(X, Y, Z)|_{Z=1}$$

- Weight enumerating function (WEF): It is a code property.
- WEF T(X) is related to IOWEF as follows

$$T(X) = T(X, Y)|_{Y=1} = T(X, Y, Z)|_{Y=Z=1}$$

Weight enumerating function calculation

- Input-output weight enumerating function (IOWEF): It is a property of the encoder.
- An alternative version of IOWEF that contains only information about input and output weights but not the length of each codeword is obtained as
$$T(X, Y) = T(X, Y, Z)|_{Z=1}$$
- Weight enumerating function (WEF): It is a code property.
- WEF T(X) is related to IOWEF as follows
$$T(X) = T(X, Y)|_{Y=1} = T(X, Y, Z)|_{Y=Z=1}$$
- Conditional weight enumerating function (CWEF): Enumerates weights of all codewords associated with particular information weights.

Weight enumerating function calculation

- For information weight of j, the CWEF is obtained as
$$T_j(X) = \sum_i A_{i,j} X^i$$
where $A_{i,j}$ represents number of code sequences with weight i, and information weight j.

Weight enumerating function calculation

- For information weight of j, the CWEF is obtained as

$$T_j(X) = \sum_i A_{i,j} X^i$$

where $A_{i,j}$ represents number of code sequences with weight i, and information weight j.

- IOWEF can be expressed in terms of CWEF as follows

$$T(X, Y) = \sum_j Y^j T_j(X)$$

Weight enumerating function calculation

- For information weight of j, the CWEF is obtained as

$$T_j(X) = \sum_i A_{i,j} X^i$$

where $A_{i,j}$ represents number of code sequences with weight i, and information weight j.

- IOWEF can be expressed in terms of CWEF as follows

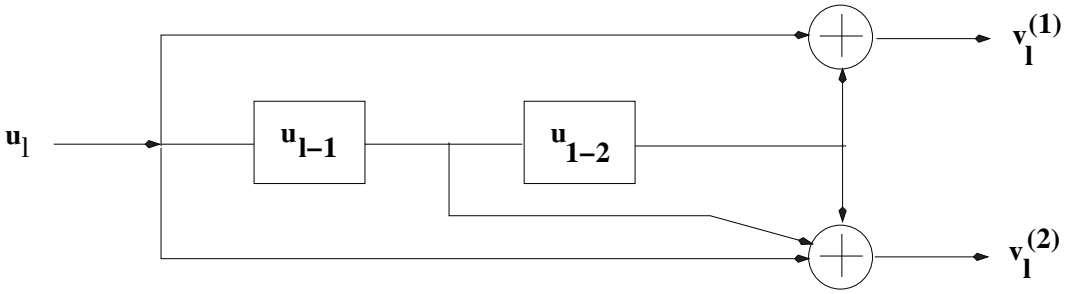
$$T(X, Y) = \sum_j Y^j T_j(X)$$

- Input-redundancy weight enumerating function (IRWEF): It is defined for systematic encoders as follows

$$T(W, Y, Z) = \sum_{w,j,l} A_{w,j,l} W^w Y^j Z^l$$

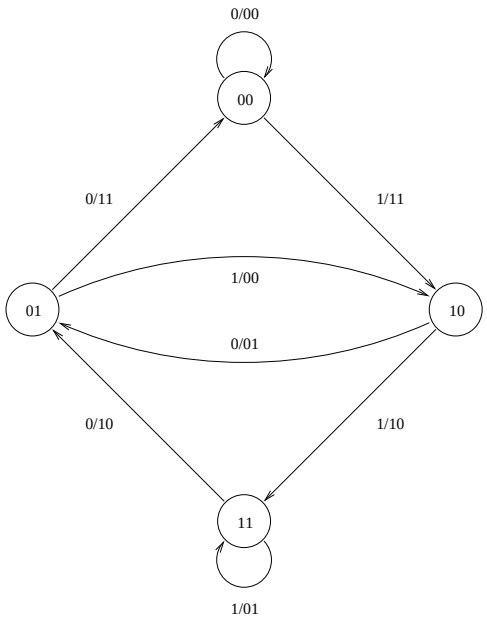
where $A_{w,j,l}$ is the number of code sequences of length l with information weight j, and parity weight w.

Convolutional encoder



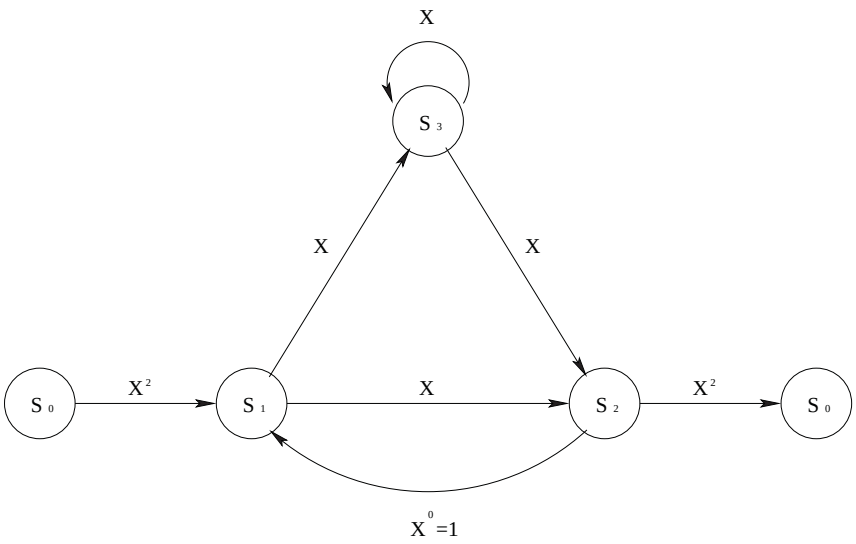
$(2, 1, 2)$ convolutional code

State diagram



State diagram of $(2, 1, 2)$ convolutional code

Modified state diagram



Modified state diagram of $(2, 1, 2)$ convolutional code

Modified State Diagram

- There are 3 loops in the modified state diagram:

Modified State Diagram

- There are 3 loops in the modified state diagram:
  - S_3S_3 with gain $C_1 = X$.

Modified State Diagram

- There are 3 loops in the modified state diagram:
  - S_3S_3 with gain $C_1 = X$.
  - $S_1S_2S_1$ with gain $C_2 = X$.

Modified State Diagram

- There are 3 loops in the modified state diagram:
  - S_3S_3 with gain $C_1 = X$.
  - $S_1S_2S_1$ with gain $C_2 = X$.
  - $S_1S_3S_2S_1$ with gain $C_3 = X^2$.

Modified State Diagram

- There are 3 loops in the modified state diagram:
  - S_3S_3 with gain $C_1 = X$.
  - $S_1S_2S_1$ with gain $C_2 = X$.
  - $S_1S_3S_2S_1$ with gain $C_3 = X^2$.
- Pairs of non-touching loops

Modified State Diagram

- There are 3 loops in the modified state diagram:
  - S_3S_3 with gain $C_1 = X$.
  - $S_1S_2S_1$ with gain $C_2 = X$.
  - $S_1S_3S_2S_1$ with gain $C_3 = X^2$.
- Pairs of non-touching loops
  - $\{1, 2\}$ with $C_1C_2 = X^2$.

Modified State Diagram

- There are 3 loops in the modified state diagram:
  - S_3S_3 with gain $C_1 = X$.
  - $S_1S_2S_1$ with gain $C_2 = X$.
  - $S_1S_3S_2S_1$ with gain $C_3 = X^2$.
- Pairs of non-touching loops
  - $\{1, 2\}$ with $C_1C_2 = X^2$.
- No triples of non-touching loops.

Modified State Diagram

- There are 3 loops in the modified state diagram:
  - S_3S_3 with gain $C_1 = X$.
  - $S_1S_2S_1$ with gain $C_2 = X$.
  - $S_1S_3S_2S_1$ with gain $C_3 = X^2$.
- Pairs of non-touching loops
  - $\{1, 2\}$ with $C_1C_2 = X^2$.
- No triples of non-touching loops.
- Hence,

$$\begin{aligned}\Delta &= 1 - (C_1 + C_2 + C_3) + C_1C_2 \\ &= 1 - 2X - X^2 + X^2 \\ &= 1 - 2X\end{aligned}$$

Weight enumerating function calculation

- There are two forward paths:

Weight enumerating function calculation

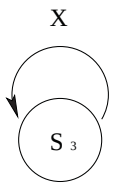
- There are two forward paths:
  - Path 1: $S_0S_1S_2S_0$ with gain $F_1 = X^5$.

Weight enumerating function calculation

- There are two forward paths:
  - Path 1: $S_0S_1S_2S_0$ with gain $F_1 = X^5$.
  - Path 2: $S_0S_1S_3S_2S_0$ with gain $F_2 = X^6$.

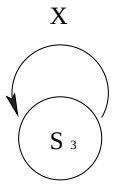
Weight enumerating function calculation

- There are two forward paths:
  - Path 1: $S_0S_1S_2S_0$ with gain $F_1 = X^5$.
  - Path 2: $S_0S_1S_3S_2S_0$ with gain $F_2 = X^6$.
- The graph G_1 obtained by removing the states on the forward path 1 is shown below.



Weight enumerating function calculation

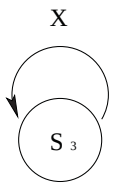
- There are two forward paths:
  - Path 1: $S_0S_1S_2S_0$ with gain $F_1 = X^5$.
  - Path 2: $S_0S_1S_3S_2S_0$ with gain $F_2 = X^6$.
- The graph G_1 obtained by removing the states on the forward path 1 is shown below.



- $\Delta_1 = 1 - X$.

Weight enumerating function calculation

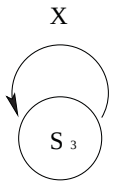
- There are two forward paths:
  - Path 1: $S_0S_1S_2S_0$ with gain $F_1 = X^5$.
  - Path 2: $S_0S_1S_3S_2S_0$ with gain $F_2 = X^6$.
- The graph G_1 obtained by removing the states on the forward path 1 is shown below.



- $\Delta_1 = 1 - X$.
- The graph G_2 obtained by removing the states on the forward path 2 is empty.

Weight enumerating function calculation

- There are two forward paths:
  - Path 1: $S_0S_1S_2S_0$ with gain $F_1 = X^5$.
  - Path 2: $S_0S_1S_3S_2S_0$ with gain $F_2 = X^6$.
- The graph G_1 obtained by removing the states on the forward path 1 is shown below.



- $\Delta_1 = 1 - X$.
- The graph G_2 obtained by removing the states on the forward path 2 is empty.
- Hence $\Delta_2 = 1$.

Weight enumerating function calculation

- The transfer function $T(X)$ is given by

$$\begin{aligned} T(X) &= \frac{F_1\Delta_1 + F_2\Delta_2}{\Delta} \\ &= \frac{X^5(1-X) + X^6 \cdot 1}{1-2X} \\ &= \frac{X^5}{1-2X} \\ &= X^5 + 2X^6 + 4X^7 + \cdots + 2^k X^{k+5} + \cdots \end{aligned}$$

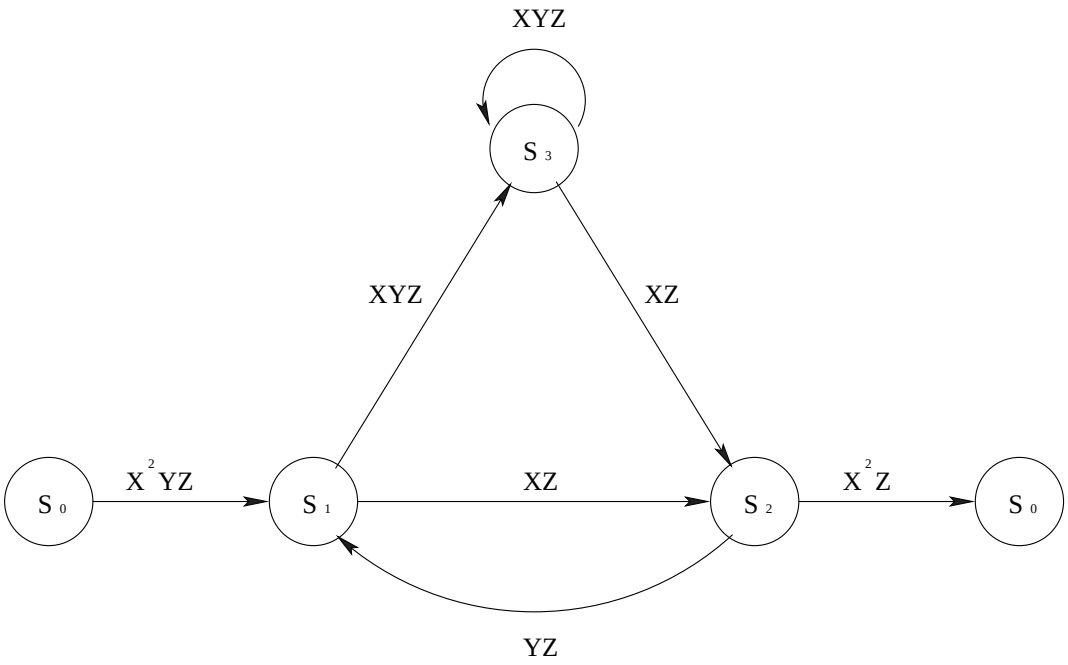
Weight enumerating function calculation

- The transfer function $T(X)$ is given by

$$\begin{aligned} T(X) &= \frac{F_1\Delta_1 + F_2\Delta_2}{\Delta} \\ &= \frac{X^5(1-X) + X^6 \cdot 1}{1-2X} \\ &= \frac{X^5}{1-2X} \\ &= X^5 + 2X^6 + 4X^7 + \cdots + 2^k X^{k+5} + \cdots \end{aligned}$$

- $d_{\text{free}} = 5$

Augmented state diagram



Input-output weight enumerating function calculation

- The transfer function $T(X, Y, Z)$ is given by

$$\begin{aligned} T(X,Y,Z) &= \frac{X^5YZ^3}{1-XYZ-XYZ^2} \\ &= X^5YZ^3 + X^6(Y^2Z^4 + Y^2Z^5) \\ &\quad + X^7(Y^3Z^5 + 2Y^3Z^6 + Y^3Z^7) + \cdots \end{aligned}$$