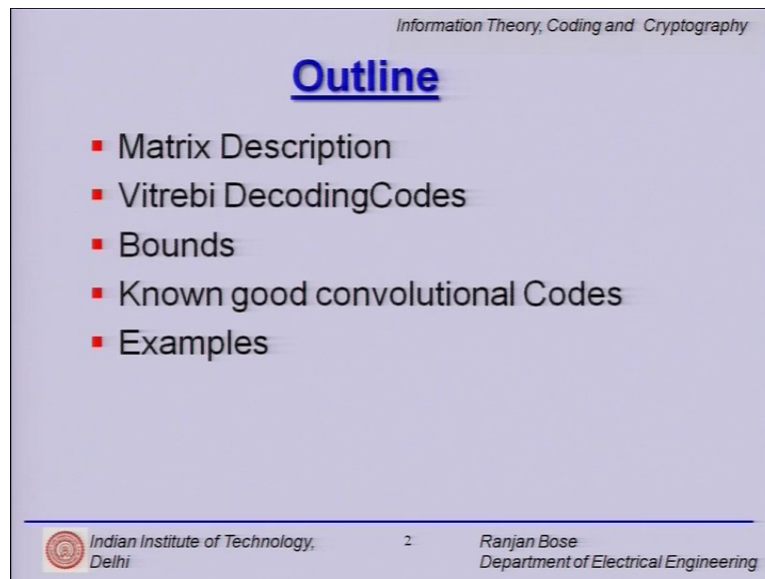


Information Theory, Coding and Cryptography
Dr. Ranjan Bose
Department of Electrical Engineering
Indian Institute of Technology, Delhi

Module – 28
Convolutional Codes
Lecture – 28

(Refer Slide Time: 00:34)



Information Theory, Coding and Cryptography

Outline

- Matrix Description
- Viterbi Decoding Codes
- Bounds
- Known good convolutional Codes
- Examples

Indian Institute of Technology, Delhi 2 Ranjan Bose
Department of Electrical Engineering


Hello and welcome to our next lecture on Convolutional Codes. Let us start with a brief outline of today's talk. We would look at the matrix description for convolutional codes. Though these are different from linear block codes, we will try to put in a matrix description. Then we will look at the decoding strategies for convolutional codes. We specifically will look at the Viterbi decoding algorithm. Then we would look at some of the bounds, both performance bounds and bounds on distance. And then we will look at some known good convolutional codes, so that is the outline for today's talk.

(Refer Slide Time: 01:08)

Information Theory, Coding and Cryptography

Recap

- Generator Polynomial Matrix
- Syndrome Polynomial Matrix
- Catastrophic and Non Catastrophic Codes
- Free Distance
- Modified State Diagram

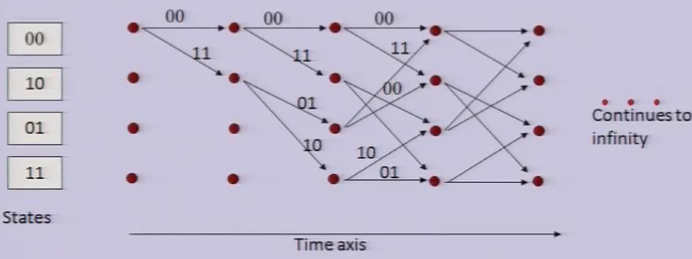
 Indian Institute of Technology, Delhi 3 Ranjan Bose
Department of Electrical Engineering

Let us start with a quick recap. We have so far studied what are convolutional codes, what is the meaning of a generator polynomial for convolutional codes. Then we looked at the syndrome polynomial matrix. We looked at catastrophic and non catastrophic convolutional codes. And we defined, the free distance of a convolutional code. We also looked at how to calculate the free distance using the modified state diagram.

(Refer Slide Time: 01:44)


Information Theory, Coding and Cryptography

Trellis Diagram



Time axis

The trellis diagram corresponding to the encoder

 Indian Institute of Technology, Delhi Ranjan Bose
Department of Electrical Engineering

So, if you remember, all convolutional codes can easily be represented in terms of a trellis diagram. If you see there are certain number of nodes, which represent the states,

so here there are four states. So, there are two memory units in this convolutional code, and they corresponds to the state 0 0, 1 0, 0 1, and 1 1.

And on the x axis, we have the time axis. So, every time an input bit comes in, it either makes a state transition or it does not. But, whatever is written on top of the branches, represent the output. An input could be either a 0 or 1, because this is an example of a 1 by 2 encoder. So, if a 1 comes in, we take the lower branch; if a 0 comes in, we take the upper branch.

So, if we have to encode 0 0 1 0, so take the upper branch, because 0, 0 then take the lower branch 1, and then take the upper branch 0. Read out what is written on top of the branches, and you have got the encoded bit stream. So, it is very easy to encode using a trellis diagram. We have also seen that any unique input bit stream corresponds to a unique path within the trellis. So, the decoding strategy should be to find out the most likely path in the trellis.

(Refer Slide Time: 03:12)

Information Theory, Coding and Cryptography


Matrix Description

- In order to obtain a generator matrix, the g_{ijl} coefficients are arranged in a matrix format.
- For each l , let G_l be a k_0 by n_0 matrix.

$$G_l = [g_{ijl}]$$
- The **generator matrix** for the convolutional code that has been truncated to a block code of blocklength n is

$$\mathbf{G}^{(n)} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \mathbf{G}_2 & \dots & \mathbf{G}_m \\ 0 & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_{m-1} \\ 0 & 0 & \mathbf{G}_0 & \dots & \mathbf{G}_{m-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{G}_0 \end{bmatrix}$$

Here $\mathbf{0}$ is a k_0 by n_0 matrix of zeros and m is the length of the shift register used to generate the code.



Indian Institute of Technology,
Delhi

Ranjan Bose
Department of Electrical Engineering

We also looked at the matrix description of convolutional codes. And what we figured out, as it is possible to have a generator k_0 matrix in terms of the g_{ijl} and l , where we have the input output relationship. So, what we can do is we can have the G_l , which is the generator matrix for the convolutional code, you can represented for a truncated block length of n , because typically a convolutional code can be thought of as having a code with infinite block lengths, because the trellis can keep going and going up to infinity.

So, what we have is a matrix description of a convolution code, which can be represented in the following format. Here, you have G_0, G_1, G_2 and we will describe them what they are up to G_m , where m is the length of the shift register used to generate the code. So, you can see there is a pattern, then there is a right shift, and then G_0 and so on and so forth, and we continue.

(Refer Slide Time: 04:26)


Information Theory, Coding and Cryptography

Matrix Description

- The generator matrix for the convolutional code is given by

$$G = \begin{bmatrix} G_0 & G_1 & G_2 & \dots & G_m & 0 & 0 & 0 & 0 & \dots \\ 0 & G_0 & G_1 & \dots & G_{m-1} & G_m & 0 & 0 & 0 & \dots \\ 0 & 0 & G_0 & \dots & G_{m-2} & G_{m-1} & G_m & 0 & 0 & \dots \\ \vdots & \vdots & & & & & & & & \ddots \end{bmatrix}$$

- The matrix extends **infinitely** far down and to the right.

 Indian Institute of Technology, Delhi
Ranjan Bose
Department of Electrical Engineering

So, for in general, when we do not restrict us to a block length of n , the generator matrix of a convolutional code can be written as follows. And the matrix practically extends up to infinity, to represent code words of infinite length.

(Refer Slide Time: 04:45)

Information Theory, Coding and Cryptography

Systematic Convolutional code

- For a systematic convolutional code, the generator matrix can be written as

$$\mathbf{G} = \begin{bmatrix}
 \mathbf{I} & \mathbf{P}_0 & 0 & \mathbf{P}_1 & 0 & \mathbf{P}_2 & \dots & 0 & \mathbf{P}_m & 0 & 0 & 0 & 0 & \dots \\
 0 & 0 & \mathbf{I} & \mathbf{P}_0 & 0 & \mathbf{P}_1 & \dots & 0 & \mathbf{P}_{m-1} & 0 & \mathbf{P}_m & 0 & 0 & \dots \\
 0 & 0 & 0 & 0 & \mathbf{I} & \mathbf{P}_0 & \dots & 0 & \mathbf{P}_{m-2} & 0 & \mathbf{P}_{m-1} & 0 & \mathbf{P}_m & \dots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & \mathbf{P}_{m-2} & 0 & \mathbf{P}_{m-1} & \dots \\
 & & & & & & & & & & \vdots & \vdots & 0 & \mathbf{P}_{m-2} & \dots \\
 & & & & & & & & & & & & \vdots & \vdots & \dots
 \end{bmatrix}$$

where \mathbf{I} is a k_0 by k_0 identity matrix, 0 is a k_0 by k_0 matrix of zeros and $\mathbf{P}_0, \mathbf{P}_2, \dots, \mathbf{P}_m$ are k_0 by $(n_0 - k_0)$ matrices.

Indian Institute of Technology, Delhi
Ranjan Bose
Department of Electrical Engineering

Similarly, if you want to write a generator matrix, which systematic for a systematic convolutional code we can write as follows. And as you can see, there are clear demarcations, which will give you this the identity matrix, and this is the parity matrix and so and so forth, you can continue right up to infinity. We will shortly look at an example, to understand how this was done.

(Refer Slide Time: 05:12)

Information Theory, Coding and Cryptography

The parity Check Matrix

- The parity check matrix can then be written as

$$\mathbf{H} = \begin{bmatrix}
 \mathbf{P}_0^T & -\mathbf{I} & & & & & & & & \dots \\
 \mathbf{P}_1^T & 0 & \mathbf{P}_0^T & -\mathbf{I} & & & & & & \dots \\
 \mathbf{P}_2^T & 0 & \mathbf{P}_1^T & 0 & \mathbf{P}_0^T & -\mathbf{I} & & & & \dots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \mathbf{P}_m^T & 0 & \mathbf{P}_{m-1}^T & 0 & \mathbf{P}_{m-2}^T & 0 & \dots & \mathbf{P}_0^T & -\mathbf{I} & \dots \\
 & & \mathbf{P}_m^T & 0 & \mathbf{P}_{m-1}^T & 0 & \dots & & & \dots \\
 & & & & \mathbf{P}_m^T & 0 & \dots & & & \dots
 \end{bmatrix}$$

Indian Institute of Technology, Delhi
Ranjan Bose
Department of Electrical Engineering

And once we have this nice looking generator matrix, which is in the echelon form, then we can easily write the parity check matrix by using the P naught transpose minus I in

the first row, and then P 1 transpose and so and so forth in the next one to have the parity check matrix.

(Refer Slide Time: 05:35)

Information Theory, Coding and Cryptography

Example

■ The generator polynomial matrix is obtained as

$$G(D) = \begin{bmatrix} D+D^2 & D^2 & D+D^2 \\ D^2 & D & D \end{bmatrix}$$

Indian Institute of Technology, Delhi
 Ranjan Bose
Department of Electrical Engineering

Let us look at a quick example. So, here we have a 2 by 3 convolution encoder, if you can see there i_1 and i_2 are the 2 bits, which are input. And we have c_1 , c_2 and c_3 are the 3 output bits. So, for every 2 bits that come in, we have 3 bits going down. So, the generator polynomial matrix should be a 2 cross 3 matrix. And if you see, how c_1 is linked to i_1 , you can see that c_1 comes as a summation of this current and a previous 1.

So, you have this D for the first delay element, and D^2 second delay element they added together right. So, you have in the first link between the first G_{11} is D plus D^2 . Similarly, c_2 is linked with i_1 only via the 2 delay unit, so D^2 . So, here is a D^2 , so you have c_2 linked with i_1 . So, this second entry is G_{12} is nothing but D^2 and so and so forth. So, since c_2 is also connected to i_2 .

You will have a G_{22} also entry here, since all are connected there are no 0s. So, we have learned how to write the generator polynomial matrix. Given any circuit for a conventional encoder, we can quickly write down the generator polynomial matrix in this case. But, from here, we will go to the generator matrix.

(Refer Slide Time: 07:21)

Information Theory, Coding and Cryptography

Example

$$G(D) = \begin{bmatrix} D+D^2 & D^2 & D+D^2 \\ D^2 & D & D \end{bmatrix}$$

- The generator polynomials are
 - $g_{11}(D) = D + D^2$,
 - $g_{12}(D) = D^2$,
 - $g_{13}(D) = D + D^2$,
 - $g_{21}(D) = D^2$,
 - $g_{22}(D) = D$and
 - $g_{23}(D) = D$.

Indian Institute of Technology, Delhi Ranjan Bose
Department of Electrical Engineering

So, if you see that we have the g_{11} as D plus D squared, g_{12} as D squared and so and so forth up to g_{23} as D . So, we have written them down right here.

(Refer Slide Time: 07:33)

Information Theory, Coding and Cryptography

Example

- To write out the matrix \mathbf{G}_0 , we look at the constants (coefficients of D^0) in the generator polynomials.
- Since there are no constant terms in any of the generator polynomials,

$$\mathbf{G}_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- Next, to write out the matrix \mathbf{G}_1 , we look at the coefficients of D^1 in the generator polynomials.
- The 1st row, 1st column entry of the matrix \mathbf{G}_1 corresponds to the coefficients of D^1 in $g_{11}(D)$.
- The 1st row, 2nd column entry corresponds to the coefficients of D^1 in $g_{12}(D)$, and so on.
- Thus,

Indian Institute of Technology, Delhi Ranjan Bose
Department of Electrical Engineering

And we now look at those matrix \mathbf{G}_0 , and we look at the constants the coefficients of D^0 in the generator polynomial. Since, there are no constant terms in any of the generator polynomials, the \mathbf{G}_0 is nothing but all 0 matrix.

(Refer Slide Time: 07:51)


Information Theory, Coding and Cryptography

Example

$$\mathbf{G}_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

▪ Similarly, we can write

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

 Indian Institute of Technology,
DelhiRanjan Bose
Department of Electrical Engineering

Now, we want to write G_1 . Please note, what we are trying to do is write these coefficients here. So, we are trying to figure out this G_0, G_1, G_2 and so and so forth up to G_m . Once we have that they nothing but right shifted versions. And please note, each one of them is a matrix. So, what we have done is we have figured out what is G_0 . Now, if you look at G_1 , we are looking at the coefficients of D^1 in the generator polynomials. So, 1st row, 1st column entry of matrix G_1 corresponds to the coefficient D^1 in g_1 .

So, we can go back, and we look at there is 1, so there is D^1 , so coefficient is 1. And similarly, we go ahead and look at it. If for g_2 , we look at the 1st row, 2nd column entry corresponds to the coefficients of D^1 and so and so forth. So, if you look at it, you see that there was 1 coefficient present with D , then there was none for the 2nd D^1 , then there is a 3rd D^1 here. So, we just go back, and check here that clearly there is a D^1 , so coefficient is 1. There are no D^1 s here, there is a single D^1 here. So, the 1st row should read 1 0 1; the 2nd row should read 0 1 1, because there is no D^1 present here, so 0 1 1.

So, if you go back 1 0 1 0 1 1. Similarly, if you have to write G_2 , then you have yes it is present. So, it is 1 1 1 1 0 0, because D^2 s are not present in these two. So, 1 1 1 1 0 0, so that gives you G_2 . So, it is very easy to write G_1, G_2 and so forth up to

G m. But, please note in our case m was equal to 2, so we stop here, and we start writing our coefficients.

(Refer Slide Time: 10:07)

Information Theory, Coding and Cryptography

Example

- The generator matrix can now be written as

$$G = \begin{bmatrix} 0 & 0 & 0 & | & 1 & 0 & 1 & | & 1 & 1 & 1 & | & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & | & 0 & 1 & 1 & | & 0 & 1 & 1 & | & 0 & 0 & 0 & \dots \\ \hline & & & & 0 & 0 & 0 & | & 1 & 0 & 1 & | & 1 & 1 & 1 & \\ & & & & 0 & 0 & 0 & | & 0 & 1 & 1 & | & 0 & 1 & 1 & \\ \hline : & & & & & & & & & & & & 1 & 0 & 1 & \\ : & & & & & & & & & & & & 0 & 1 & 1 & \dots \end{bmatrix}$$

Indian Institute of Technology, Delhi Ranjan Bose
Department of Electrical Engineering

So, this is G₀, this is G₁, G₂, and then 0s continue forever. And then you start with 0 0 0 matrix. And then it is a shifted version, this is G₀, G₁, G₂, and so and so forth G₀, G₁, G₂. So, you will now have a generator matrix for the convolutional encoder.

(Refer Slide Time: 11:05)

Information Theory, Coding and Cryptography

Decoding

- There are **three** important decoding techniques for convolutional codes:
- Threshold decoding, sequential decoding and the Viterbi decoding.
- The **Sequential Decoding** technique was proposed by Wozencraft in 1957.
- Sequential decoding has the advantage that it can perform very well with long-constraint-length convolutional codes, but it has a variable decoding time.
- Threshold Decoding**, also known as **Majority Logic Decoding**, was proposed by Massey in 1963 as a part of his doctoral thesis at MIT.
- Threshold decoders were the first commercially produced decoders for convolutional codes.
- Viterbi decoding** was developed by Andrew J. Viterbi in 1967.

Indian Institute of Technology, Delhi Ranjan Bose
Department of Electrical Engineering

Now, we come to the most important part of convolutional code, which is the decoding part. We have seen that the encoding is very easy, very hardware friendly, extremely easy

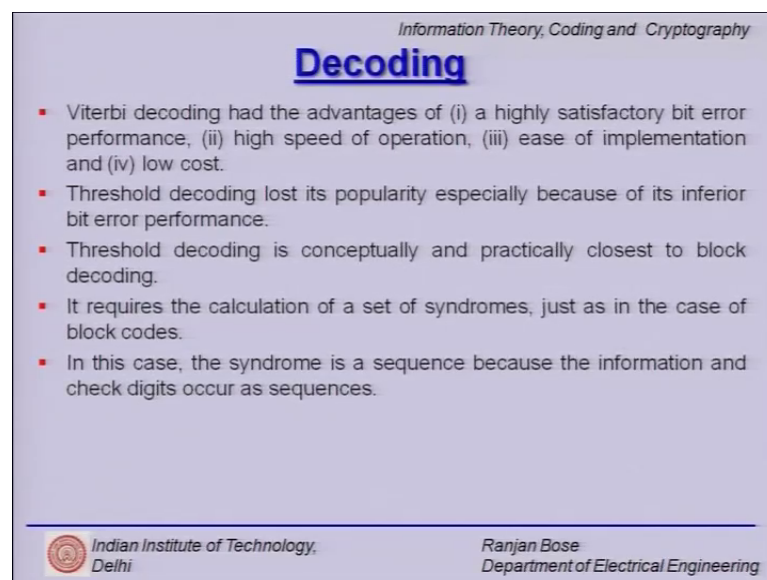
to perform in terms of a trellis code. So, the computational complexity is minimal at the encoding side. But, if you look at the decoding, we have a little bit of difficulty, it is a slightly involved process as we will figure out.

So, there are three important decoding techniques for convolutional codes. Threshold decoding, sequential decoding, and the most famous Viterbi decoding.

The sequential decoding technique was originally proposed by Wozencraft, and 1957. And it has an advantage that it can perform very well with long-constraint-length convolutional code, but it has a variable decoding time. So, it is not so suited for real time operations.

The second technique, which is the threshold decoding is also known as the Majority Logic Decoding, was proposed by Massey 1963; And the threshold decoders of first commercially produced decoders for convolutional codes, so they had a lot of merit. But, the optimal Viterbi decoding developed by Andrew Viterbi in 1967. Really has been the mainstay has been the choice best decoding technique, which was used in the industry today.

(Refer Slide Time: 12:06)



The slide is titled "Decoding" and is part of a presentation on "Information Theory, Coding and Cryptography". It lists five bullet points comparing Viterbi decoding and threshold decoding. At the bottom, it identifies the speaker as Ranjan Bose from the Department of Electrical Engineering at the Indian Institute of Technology, Delhi.

Information Theory, Coding and Cryptography

Decoding

- Viterbi decoding had the advantages of (i) a highly satisfactory bit error performance, (ii) high speed of operation, (iii) ease of implementation and (iv) low cost.
- Threshold decoding lost its popularity especially because of its inferior bit error performance.
- Threshold decoding is conceptually and practically closest to block decoding.
- It requires the calculation of a set of syndromes, just as in the case of block codes.
- In this case, the syndrome is a sequence because the information and check digits occur as sequences.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So, why is Viterbi decoding so popular well it has some very good advantages, a highly satisfactory bit error rate performance, very high speed of operation, very easy hardware friendly to implement, and low cost. So, these are the reasons why Viterbi coding has


become very very popular. The threshold decoding in comparison to Viterbi decoding had an inferior bit error rate performance. So, the threshold decoding is closes practically to block decoding. And it requires the calculation of syndromes. And syndrome is a sequence, because the information and the check bits occur at as sequences. So, the syndrome is a sequence.

(Refer Slide Time: 12:57)

Information Theory, Coding and Cryptography

Viterbi Decoding

- Viterbi decoding has the advantage that it has a **fixed decoding time**.
- It is well suited to **hardware** decoder implementation.
- But its **computational requirements** grow exponentially as a function of the constraint length, so it is usually limited in practice to constraint lengths of $v = 15$ or less.

 Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering


But let us now focus on the Viterbi decoding algorithm, which is a fixed time decoding algorithm. So, the decoding time does not really vary in terms of the input that is coming in. It is also extremely hardware friendly, it can be easily put on (Refer Time: 13:17), but, its computational requirement is high. So, there is no free lunch, and it grows exponentially as a function of the constraint length. So, in real life, the constraint length of 15 or less is typically, the ones for which Viterbi decoding is popularly used.

(Refer Slide Time: 13:35)

Information Theory, Coding and Cryptography

Optimum Decoding

- Optimum decoding will result in a minimum probability of decoding error.
- Let $p(r|c)$ be the **conditional probability** of receiving r given that c was sent.
- We can state that the optimum decoder is the maximum likelihood decoder with a decision rule to choose the code vector estimate \hat{c} for which the log-likelihood function $\ln p(r|c)$ is maximum.
- If we consider a binary symmetric channel where the vector elements of c and r are denoted by c_i and r_i , then we have
$$p(\mathbf{r} | \mathbf{c}) = \prod_{i=1}^N p(r_i | c_i)$$
where N is the length of the sequence.
- Hence the log-likelihood function equals
$$\ln p(\mathbf{r} | \mathbf{c}) = \sum_{i=1}^N \ln p(r_i | c_i)$$

 Indian Institute of Technology,
DelhiRanjan Bose
Department of Electrical Engineering

So, let us spend some time looking at this optimal decoding strategy. And we will spend a few slides looking at how and why Viterbi decoding does an optimal decoding. So, this optimal decoding technique results in a minimum probability of error. So, let us say the probability of r giving c , where r was the received vector, and c was what was sent, so that will be the conditional probability of receiving r given c was sent.

And we can see that the optimal decoder is the maximum likelihood decoder with the decision rule to choose the code vector estimate \hat{c} for which the log-likelihood function $\ln p(r|c)$ is maximum. See we want to maximize this probability of r being received given c was sent. And we would like to use the log-likelihood function. So, if you consider a binary symmetric channel, where the vector elements of c and r are denoted by c_i and r_i , because if we if you remember, we are decoding a long sequence, and this is encoded using a trellis diagram.

So, for the entire chain, so $p(r|c)$ is nothing but the product of $p(r_i|c_i)$, and n is the length of the sequence. This product necessitates the use of a log function in front, and we have this log-likelihood function, $\ln p(r|c)$, and the log converts a product into an addition, so that makes life simpler for us.

(Refer Slide Time: 15:30)


Information Theory, Coding and Cryptography

Optimum Decoding

- Let us assume

$$p(r_i | c_i) = \begin{cases} p, & r_i \neq c_i \\ 1-p, & r_i = c_i \end{cases}$$
- If we suppose that the received vector differs from the transmitted vector in **exactly d positions** (the Hamming distance between vectors \mathbf{c} and \mathbf{r}), we may rewrite the log-likelihood function as

$$\begin{aligned} \ln p(\mathbf{r} | \mathbf{c}) &= d \ln p + (N-d) \ln(1-p) \\ &= d \ln \left(\frac{p}{1-p} \right) + N \ln(1-p) \end{aligned}$$

 Indian Institute of Technology, Delhi
 Ranjan Bose
Department of Electrical Engineering

So, let us assume that r_i is received given c_i was sent the probability is p , r_i not equal to c_i , so that is the probability of error, and $1 - p$ and r_i is equal to c_i . So, p stands for the probability of error. And suppose the received vector differs from the transmitted vector at exactly d positions ok, so there are d errors that have happened. Then we can simply plug in, and the log-likelihood function, now looks like $\ln p(\mathbf{r} | \mathbf{c})$ as a sequence of received bits, and \mathbf{c} is the sequence of transmitted bits, it just comes out to be as follows. And this is coming directly from this log-likelihood function, and you can write it as follows.


(Refer Slide Time: 16:29)

Information Theory, Coding and Cryptography

Optimum Decoding

- We can assume the probability of error $p < \frac{1}{2}$ and we note that $N \ln(1-p)$ is a constant for all code vectors.
- Now we can make the statement that the maximum likelihood decoding rule for a binary symmetric channel is to choose the code vector estimate $\hat{\mathbf{c}}$ **that minimizes the Hamming distance between the received vector \mathbf{r} and the transmitted vector \mathbf{c} .**
- For soft decision decoding in additive white Gaussian noise (AWGN) channel with single sided noise power N_0 , the likelihood function is given by

$$\begin{aligned} p(\mathbf{r} | \mathbf{c}) &= \prod_{i=1}^N \frac{1}{\sqrt{\pi N_0}} e^{-\frac{|r_i - c_i|^2}{N_0}} \\ &= \left(\frac{1}{\pi N_0} \right)^{\frac{N}{2}} \exp \left(-\frac{1}{N_0} \sum_{i=1}^N |r_i - c_i|^2 \right) \end{aligned}$$

 Indian Institute of Technology, Delhi
 Ranjan Bose
Department of Electrical Engineering

So, we assume that probability of error p is less than half right. And we also known as at $N \log 1$ minus p is a constant for all code vectors. So, now, we can make a statement that the maximum likelihood decoding rule for binary symmetric channel is to choose the vector estimate \hat{c} , which minimizes the hamming distance between the received vector r , and the transmitted vector c . So, our aim is to have a decoding strategy that minimizes this hamming distance.

So, for soft decision decoding in additive white Gaussian noise with single sided noise power N naught, the likelihood function can alternatively be written as follows. So, this we have studied in our communication theory course. But, so far for hard decision decoding with d errors, we have already derived the expression.

(Refer Slide Time: 17:26)


Information Theory, Coding and Cryptography

Optimum Decoding

- Thus the maximum likelihood decoding rule for the AWGN channel with soft decision decoding is to minimize the squared Euclidean distance between r and c .
- This squared Euclidean distance is given by

$$d_E^2(\mathbf{r} | \mathbf{c}) = \sum_{i=1}^N |r_i - c_i|^2$$

- Viterbi decoding works by choosing that trial information sequence, the encoded version of which is **closest** to the received sequence.
- Here, Hamming distance will be used as a measure of closeness between two sequences.



Indian Institute of Technology,
Delhi

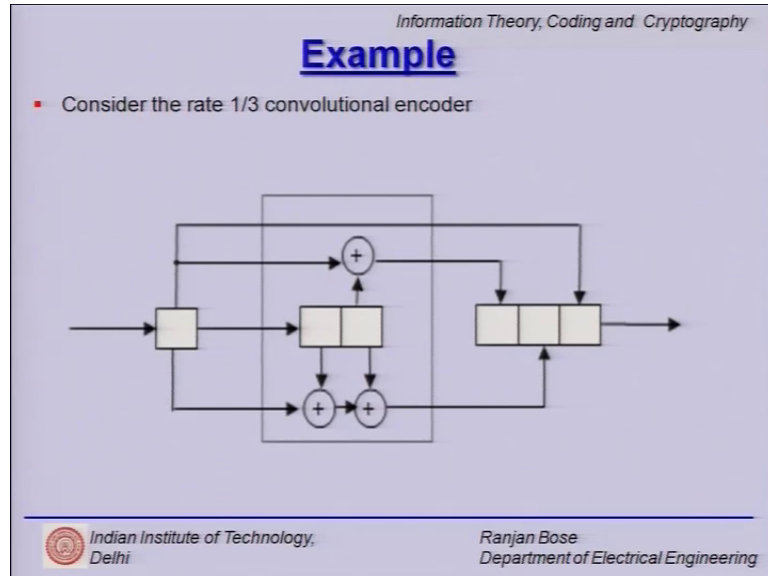
Ranjan Bose
Department of Electrical Engineering

So, the maximum likelihood decoding rule for Additive White Gaussian channel right. With soft decision decoding is to minimize the squared Euclidean distance between r and c ; And the Euclidean distances given as follows. So, the Viterbi decoding works by choosing that trial information sequence, where the encoded version is closest to the received sequence.

So, if you are using hard decision decoding, then it is in terms of the hamming distance, if you are using soft decision decoding, we are using in terms of the Euclidean distance, squared Euclidean distance. So, for hard decision decoding, will be using hamming distance as a measure of closeness between the two sequence. So, when we say, we are

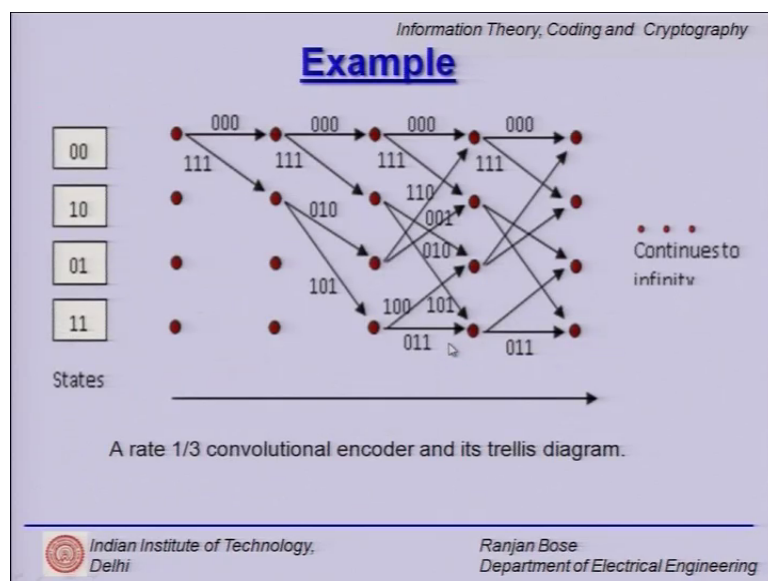
looking at a decoding strategy to pick out the most likely sequence that was sent, then we are finding out the closest sequence with respect to the received sequence.

(Refer Slide Time: 18:29)



Let us understand this using a simple example. So, suppose we have a rate 1 by 3 convolution encoder. So, you can see that the input is a single bit, and output are 3 bits. And the number of states in the trellis should be 4, because there are two elements in the memory.

(Refer Slide Time: 18:54)



So, constraint length is 2. So, here we have a Viterbi decoding example, where there are four states. And if you see on top of each branch, there are 3 bits written, simply because the output is 3 bits for every single bit comes that comes in. So, this input bit can be a white that a 0 or a 1. So, in the trellis diagram, you have two branches coming out from every node, 0 means take the upper branch, 1 means take the lower branch.

So, suppose we have to encode 1 0 0 1, so 1 means take the lower branch, 0 means take the upper branch, 0 means take the upper branch, and 1 means take the lower branch. So, you read out what is written on top of the branches, this is the rate 1 by 3 convolutional encoder. We will use this as an example to demonstrate how Viterbi decoding is done, in terms of the optimal decoding strategy that we worked out.

(Refer Slide Time: 19:57)


Information Theory, Coding and Cryptography

Example

- Suppose the transmitted sequence was the all zero sequence.
- Let the received sequence be

$$r = 010000100001\dots$$
- Since it is a 1/3 rate encoder, we first segment the received sequence in groups of three bits (because $n_0 = 3$), i.e.,

$$r = 010\ 000\ 100\ 001\ \dots$$
- The task at hand is to find out the most likely path through the trellis.
- Since a path must pass through nodes in the trellis, we will try to find out which nodes in the trellis belong to the most likely path.



Indian Institute of Technology,
Delhi

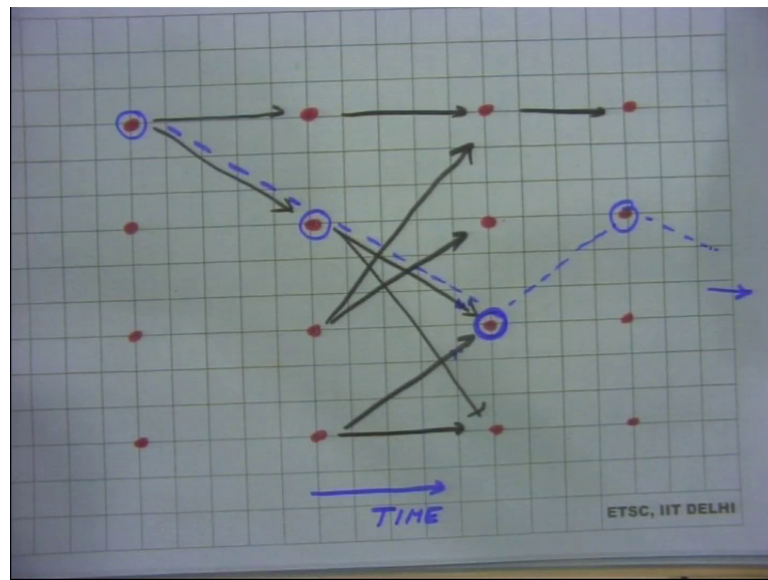
Ranjan Bose
Department of Electrical Engineering

So, without loss in generality, let us say the all zero sequence was sent right. But, what we have received is a sequence with a few errors, it is very clear easy to see that the ones denote, where there is have been seen. So, we do not know at the receiver what was sent, but at the transmitter suppose the all 0 sequence was sent. Congressional encoders are linear, so we can work this example out with respect to the all 0 sequence, and we can generalize this example.

So, this is a rate 1 by 3 encoder, so we work in groups of 3. So, first 3 bits is processed as stage 1, then the next 3 bits, next 3 bits and so and so forth, simply because n naught is 3. So, we just take this long receive sequence, and subdivide it into groups of 3, so 3, then

next 3, next 3, and next 3. Now, the objective is to find out the most likely path through the trellis that resulted in this one right. Since, the path must pass through nodes in the trellis, we will try to find out, which nodes in the trellis belongs to the most likely path in the trellis.

(Refer Slide Time: 21:33)



So, let us quickly understand this rational by looking at a trellis here. So, if you see, we have nodes in the trellis. And since, the number of states is four, we have four nodes at every stage, and this continues up to infinity.

Now, clearly there are paths in the trellis, which corresponds to an input sequence. And some sequence has been transmitted, which results in a particular path, so we do not know, which one is it, but it will keep going and so and so forth, and this continues up to infinity. The problem is that as we go along the time axis, the number of path grows exponentially, this is seen, because if we have to match and find out, which is the most likely path.

Then we have to keep track, save in memory all the paths compare with each and every path, the hamming distance and find out, which is the most likely path. So, what the Viterbi algorithm does is it makes a very simple observation. It says the correct path must pass through the correct nodes. So, even though the paths may grow exponentially, the number of nodes are fixed, there are only four nodes at every stage.

So, instead of keeping track of each and every path, we try to figure out what are the most likely nodes from which this optimal path is passing, and we keep a track of nodes. And how do we do that, how do we prevent the number of nodes growing exponentially well. We have to stick to the number of nodes here, so if you see, if you look at this particular node, there are two paths that are coming through this one. Now, only one of the path is correct, so at this node, we will take a call as to which path is more likely. If it is an optimal path, then it will be optimal at this point, so you can always choose the better path and prune this tree.

So, at any stage, if you look at a node, we can take a decision at the node. And choose and retain, the most likely path through that point, and throw away all the other paths. So, we discard. So, we keep pruning this and M-ary tree and so. The exponential growth that is expected of the paths is not seen, because we just keep a track of which of the nodes, which are optimal. So, this is the most critical observation, which helps us keep the whole mathematical operation of decoding tractable, and we do not run out of memory, because we do not grow exponentially.


So, we now go back to our slides and try to see how this example illustrates this point. So, we go back and see, and make this observation. Since, a path must pass through the nodes in the trellis, we will try to find out which nodes in the trellis belong to the most likely path right. This is the key observation.

(Refer Slide Time: 25:23)

Information Theory, Coding and Cryptography

Example

- At any time, every node has two incoming branches (\rightarrow).
- We simply determine which of these two branches belongs to a more likely path (and discard the other).
- We take this decision based on some metric (Hamming distance).
- In this way we just have to retain just one path per node and the metric of that path.
- In this example, we will have to retain only four paths as we progress with our decoding (since we have only 4 states in our trellis).

 *Indian Institute of Technology,
Delhi*

*Ranjan Bose
Department of Electrical Engineering*

So, you see at every node you have two incoming branches which simply determines, which of these two branches belongs to a more likely path, and discard the other. This process of discarding paths at every stage, keeps in check the number of paths, because if their path was suboptimal at that stage, there is no way it will become optimal later ok; this is a principle of optimality.


So, we take this decision based on some metric, and that metric, we know for hard decision decoding is the hamming distance. In this way, we have to retain just one path per node, and the metric of that path ok. So, in this example, we will have to retain only four paths as we progress with that decoding. Since, we have only four states in our trellis, and it will be four only, it will never grow.

(Refer Slide Time: 26:20)

Information Theory, Coding and Cryptography

Example

- Let us consider the first branch of the trellis which is labeled 000.
- We find the Hamming distance between this branch and the first received framelength, 010.
- The Hamming distance $d(000, 010) = 1$.
- Thus the metric for this first branch is 1, and is called the **branch metric**.
- Upon reaching the top node from the starting node, this branch has accumulated a metric = 1.
- Next we compare the received framelength with the lower branch, which terminates at the second node from the top.

 Indian Institute of Technology,
DelhiRanjan Bose
Department of Electrical Engineering

So, let us consider the first branch of the trellis, which is labeled 0 0 0. And we find the hamming distance between this branch, and the first received frame, but if you remember, we received 0 1 0 as the first 3 bits. So, we measure that the hamming distance with this path, and all 0 path you get 1. This is your branch metric. Upon reaching the top node from the starting node, this branch accumulates a metric of 1 unit. But, we have another option, so we looked at the next branch. So, we compare the received frame with the lower branch, the terminates on the second node from the top


(Refer Slide Time: 27:00)

Information Theory, Coding and Cryptography

Example

- The Hamming distance in this case is $d(111, 010) = 2$.
- Thus the metric for this first branch is 2.
- At each node we write the total metric accumulated by the path, called the **path metric**.
- The path metrics are marked by circled numbers in the trellis diagram.
- At the subsequent stages of decoding when two paths terminate at every node, we will retain the path with the smaller value of the metric.

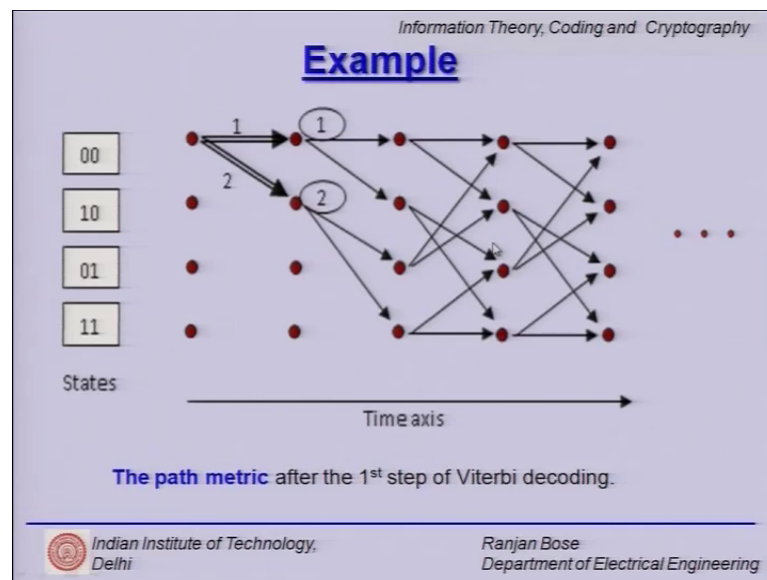
↘

 Indian Institute of Technology,
Delhi

Ranjan Bose
Department of Electrical Engineering

Here, it is 1 1 1, so the distance between 1 1 1, and 0 1 0 is 2, so that is the branch metric for that one.

(Refer Slide Time: 27:10)

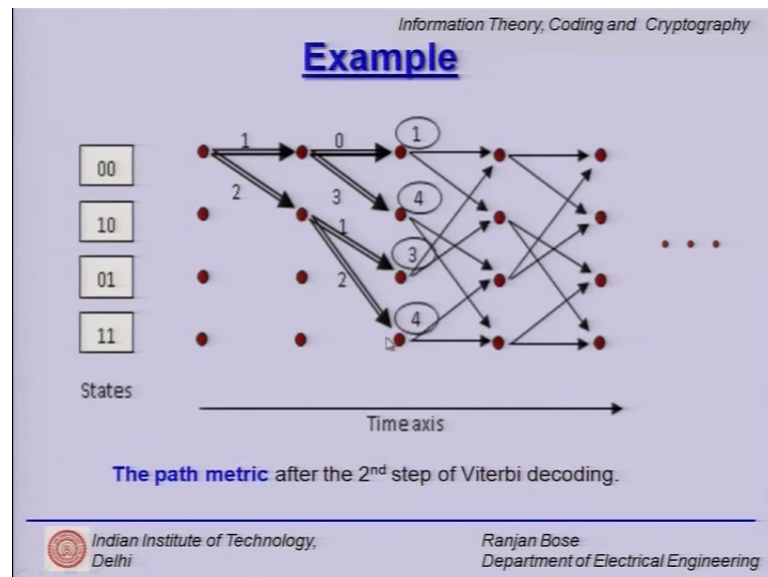


So, if you go back to a trellis, what we have done here is we started always from the base 0 0. We compared the received vector with the first branch here, and we founded a hamming distance of 1, because it was labeled 0 0 0. The second this branch was labeled 1 1 1, we compared our hamming distance of the received branch, and what is written on

this 1, and we got a distance of 2. So, we write here that for the two paths that we have progressed through, we have a branch metric 1 and 2 and a path metric of 1 and 2 here.

But, if you know we can keep continuing to the next step, and then again find out the hamming distance between the second frame received, and what is written on top of this branch. Second frame received compare it with this branch, second frame received compare it with this branch, and second frame received and compare it with this branch. So, again we will have four branch matrix here, and we add it up to yield the path matrix. So, we will have path metric four of them written here.

(Refer Slide Time: 28:23)

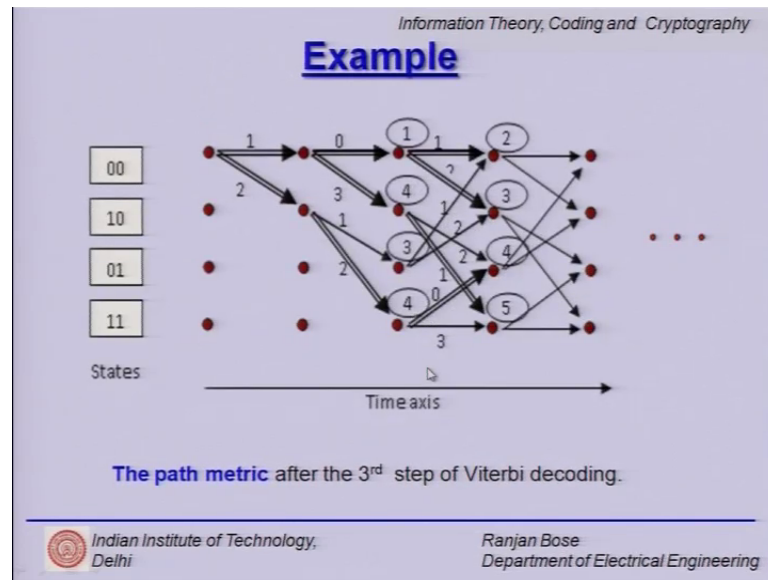


So, let us proceed here. And what we find is that the 2nd frame received is 0 0 0, and what is written on this branch is also 0 0 0. So, between the received, and what was written on the top of the trellis, there is 0 additional hamming distance. So, the branch metric is 0, but already there was a burden of 1, so 1 plus 0 the total branch matrix for the two add up, and give you the path metric of 1. We do not stop here.

We look at the second possibility, because anybody could be a winner. So, 1, and if this branch was sent, you have this return here, but this lower branch has written on up on top of it 1 1 1 in the trellis diagram. But, what we have received in the 2nd frame is 0 0 0, so we have a branch metric of 3 leading to 3 plus 1 path metric of 4.

Similarly, if you look at this 1, here you compare it, and you get an additional hamming bit of 1 here, so 2 plus 1 3 and 2 plus 2 4. So, at the end of two steps of the Viterbi decoding encircled are the path matrix for the four cases, branch 1 leading to path 1, path 2, path 3, and path 4, but we cannot declare the results yet. So, we keep going.

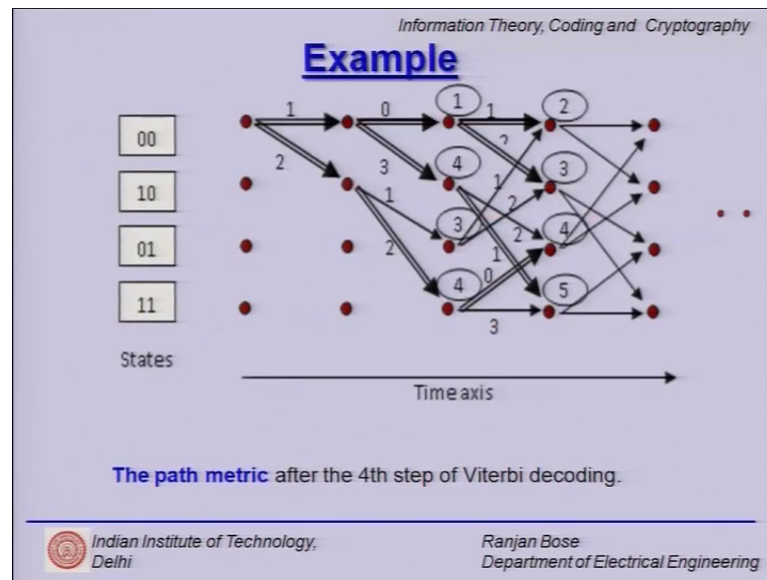
(Refer Slide Time: 29:54)



And we look at the 3rd Viterbi algorithms decoding stage, and we compute further matrix for the branch. So, what you receive again is 1 0 0 1, and here what is written on top is 0 0 0. So, the branch metric is 1, path metric was already 1. So, path metric plus branch matrix give you 2 ok. Similarly, you can do for all four, and you can keep adding. What is interesting is now two branches are converging at every node. Each of these paths, which are converging at every node has its path metric with it.

Now, we simply discard the 1 with a larger path matrix, because it is already suboptimal at that stage. And we retain the 1, which is having a lower path metric. So, we discard. So, we do the pruning act for every branch and every node here. So, consequently, we have written down the minimum branches and minimum path matrix here.

(Refer Slide Time: 31:21)



So, at the end of 4th stage, we repeat this operation, and we keep doing it here. So, we have at the end of 4th stage a path metric of 2, 3, 4, and 5, clearly we can see that this first branch has accumulated minimum hamming distance, and is the most likely winner up to this point ok. So, what it says is we have received a certain sequence.

Suppose this all 0 sequence was sent, then the hamming distance is 2. Suppose this 0 0 0 0 0 1 1 1 was sent, then the hamming distance sequence to sequence would be 3. Similarly, if you look at these this 1, it will be 4. So, we have already highlighted the 1 with a minimum path metric. And suppose you look at this path, then it will be 5. So, we as we have 4 choices, but we have already know, which 1 appears to be the winner.

(Refer Slide Time: 32:25)


Information Theory, Coding and Cryptography

Example

- The minimum distance for this code is $d^* = 6$.
- The number of errors that it can correct per framelength is equal to

$$t = \lfloor (d^* - 1)/2 \rfloor = \lfloor (6 - 1)/2 \rfloor = 2.$$

- In this example, the maximum number of errors per framelength was 1.

 Indian Institute of Technology,
DelhiRanjan Bose
Department of Electrical Engineering


So, the minimum distance of this code d^* is 6. So, the number of errors it can write per frame is equal to $d^* - 1$ by 2, this is the smallest integer less than or equal to, so that is 2. So, in this example, the maximum number of errors per frame was 1, so it could be corrected.

(Refer Slide Time: 32:57)

Information Theory, Coding and Cryptography

Example

- Consider the set of surviving paths at the i^{th} frame time.
- If all the **surviving paths** cross through the same nodes then a decision regarding the most likely path transmitted can be made up to the point where the nodes are common.
- To build a practical Viterbi decoder, one must choose a decoding window width w , which is usually several times as big as the blocklength.
- At a given frame time, f , the decoder examines all the surviving paths to see if they agree in the first branch.
- This branch defines a decoded information frame and is passed out of the decoder.

 Indian Institute of Technology,
DelhiRanjan Bose
Department of Electrical Engineering

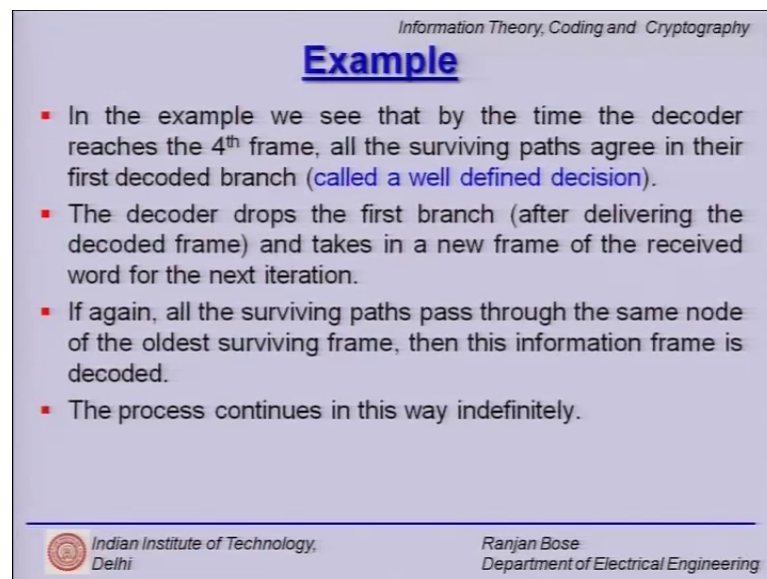
And we look at the surviving paths for first case, and we say that if all the surviving paths cross the same nodes, then a decision regarding the most likely transmitted path can be made up to the point, where the nodes are common. So, please note, we have

introduced this notion of discarding paths, and whatever remains are the surviving paths. So, you can choose a time, when you can declare the results, and we will show it by a decoding window.

So, to build a practical Viterbi decoder, because we can keep going on and on, but one must choose a decoding window of width w ; after which we will say, enough is enough, time to declare the result. So, at a given time frame f , the decoder examines all the surviving pass to see, if they agree in the first branch. This branch defines a decoded information frame, and is passed out to the decoder.

So, if you see in the previous example, it is too early to declare the result, but if you see that these two are the most likely paths, and they agree in the first two branches. So, you can start declaring the results, or you can wait till the end of the decoding time frame, and then declare the results.

(Refer Slide Time: 34:31)



Information Theory, Coding and Cryptography

Example

- In the example we see that by the time the decoder reaches the 4th frame, all the surviving paths agree in their first decoded branch (called a well defined decision).
- The decoder drops the first branch (after delivering the decoded frame) and takes in a new frame of the received word for the next iteration.
- If again, all the surviving paths pass through the same node of the oldest surviving frame, then this information frame is decoded.
- The process continues in this way indefinitely.

Indian Institute of Technology, Delhi

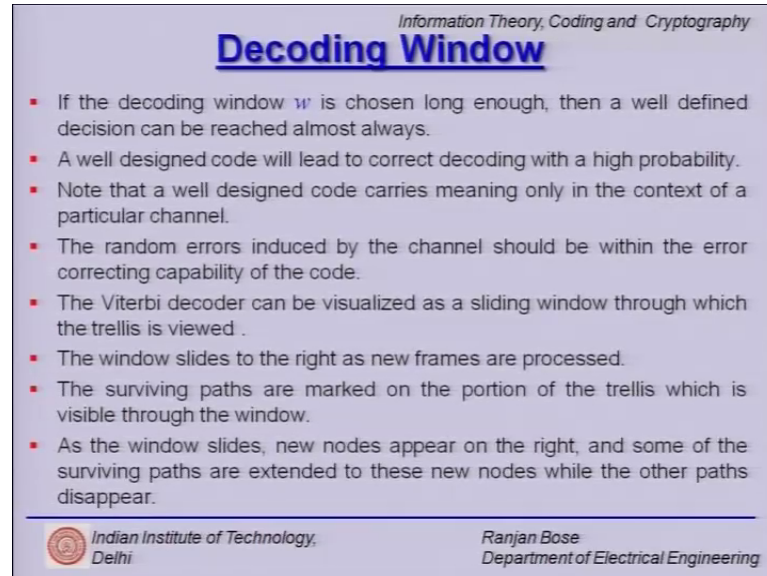
Ranjan Bose
Department of Electrical Engineering

So, in the example, we see that by the time we reach the 4th frame, all the survival pass agree in their first decoded branch, and this is called a well defined decision. See, we go back and we can see that they start agreeing in the first branch, and so we say that ok, we are pretty confident that yes that is exactly what was sent.

So, the decoder dropped the first branch, after delivering the decoded frame, and takes in a new frame of the received word for next iteration, so it slides the window. If again, all

the surviving paths pass through the same node of the oldest surviving frame, then this information frame is decoded.


(Refer Slide Time: 35:22)



Information Theory, Coding and Cryptography

Decoding Window

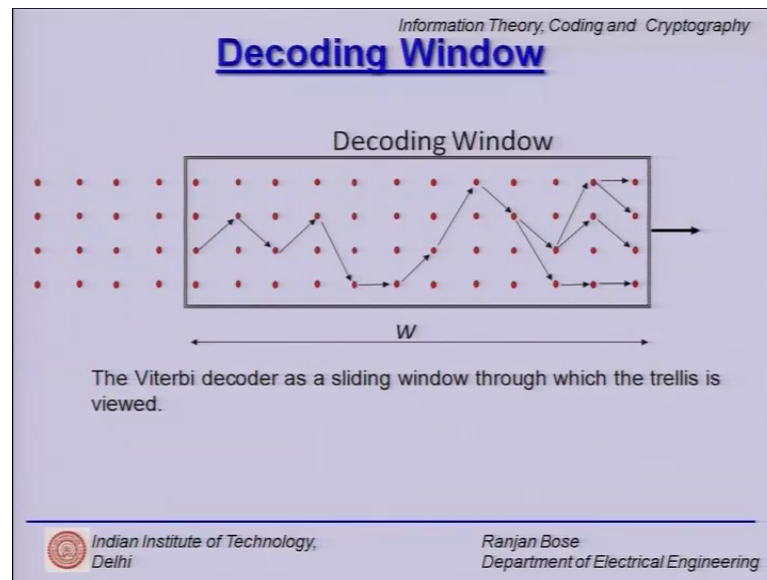
- If the decoding window w is chosen long enough, then a well defined decision can be reached almost always.
- A well designed code will lead to correct decoding with a high probability.
- Note that a well designed code carries meaning only in the context of a particular channel.
- The random errors induced by the channel should be within the error correcting capability of the code.
- The Viterbi decoder can be visualized as a sliding window through which the trellis is viewed .
- The window slides to the right as new frames are processed.
- The surviving paths are marked on the portion of the trellis which is visible through the window.
- As the window slides, new nodes appear on the right, and some of the surviving paths are extended to these new nodes while the other paths disappear.

 *Indian Institute of Technology,
Delhi* *Ranjan Bose
Department of Electrical Engineering*

So, let us talk about the decoding window now. If the decoding window w is chosen long enough, then a well defined decision almost always be reached. A well designed code will lead to current decoding with a high probability. So, we have not talked about how to design good codes, but yes if you have to maximize the d_{free} , if the d_{free} is high, then with a reasonably large decoding window, you can almost always get the right answer. So, a well designed code carries meaning only in the context of a particular channel. So, different kinds of channel may require different design constraints for your Viterbi for your convolutional encoder.

The random errors introduced by the channel should be within the error correcting capability of the code. So, the Viterbi decoder can be visualized as a sliding window, through which a trellis is viewed. And the window slides to the right as new frames are processed, and older results are declared. The surviving paths are marked on the portions of the trellis, which is visible through the window. As the window slides, new nodes appear on the right, and some of the surviving paths are extended to the new nodes, while other paths disappear.

(Refer Slide Time: 36:41)




So, this is like a visual of the decoding window. There is an semi infinite trellis, the decoding windows already passed through the first 4 frames, and it has declared the result. And it has some of the surviving paths, and it keeps moving the window of size w , and as it go moves forward, it keeps declaring the results and continues. So, that is how it can be visualized.

(Refer Slide Time: 37:07)

Information Theory, Coding and Cryptography

Errors !

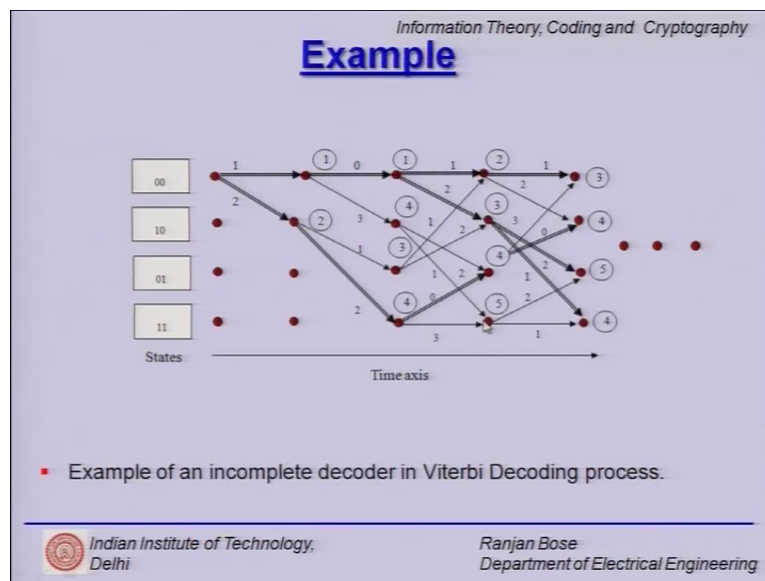
- It may be possible that in some cases the decoder reaches a well defined decision, but a wrong one!
- If this happens, the decoder has no way of knowing that it has taken a wrong decision.
- Based on this wrong decision, the decoder will take more wrong decisions.
- However, if the code is non-catastrophic, the decoder will recover from the errors.

 Indian Institute of Technology,
DelhiRanjan Bose
Department of Electrical Engineering

Now, it may be possible that in some cases, the decoder reaches a well defined decision, but a wrong one, it is possible right. If the number of errors are more than it can correct,

it will go and declare a result, which is incorrect. If this happens, the decoder has no way of knowing that it has taken a wrong decision. Now based on this wrong decision, the decoder will take more wrong decisions, so error propagation will happen. But, if the code is a non-catastrophic code, the decoder will recover from the errors. So, we have already studied catastrophic and non-catastrophic codes. So, this is the implication of that.

(Refer Slide Time: 37:46)



So, this is an example of an incomplete decoder in the Viterbi decoding process.


(Refer Slide Time: 37:55)

Information Theory, Coding and Cryptography

Distance bounds

- Upper bounds can be computed on the minimum distance of a convolutional code that has a rate $R = \frac{k_0}{n_0}$ and a constraint length $v = mk_0$.
- These bounds are similar in nature and derivation to those for block codes, with block length corresponding to constraint length.
- However, as we shall see the bounds are not very tight.
- These bounds just give us a rough idea of how good the code is.
- Here we present the bounds (without proof) for binary codes.
- For rate R and constraint length, let d be the largest integer that satisfies

$$H\left(\frac{d}{n_0 v}\right) \leq 1 - R$$

 Indian Institute of Technology, Delhi
Ranjan Bose
Department of Electrical Engineering

Now, let us spend some time looking at the distance bounds. So, we have already looked at encoder, and the decoder for convolutional codes. Now, we have already talked about d_{free} . And then quickly let us look at some of the bounds on this d_{free} or minimum distance of the codes. So, upper bounds can be computed on the minimum distance of a convolutional code that is a rate R is equal to k/n , and a constraint length n_0 equal to m .

These bounds are similar in nature, and derivation to those for block codes, but some of the bounds are not really tight, but still it gives us a fair idea right. So, for a rate R and constraint length n_0 , let d_{min} with the largest integer that satisfies H , which is the entropy function, d_{min}/n_0 that is less than or equal to $1 - H(R)$.

(Refer Slide Time: 39:00)

Information Theory, Coding and Cryptography

Distance bounds


- Then at least one binary convolutional code exists with minimum distance d_{min} for which the above inequality holds.
- Here $H(x)$ is the familiar entropy function for a binary alphabet

$$H(x) = -x \log_2 x - (1-x) \log_2 (1-x), \quad 0 \leq x \leq 1.$$

- For a binary code with $R = 1/n_0$ the minimum distance d_{min} satisfies

$$d_{\text{min}} \leq \lfloor (n_0 v + n_0) / 2 \rfloor,$$

- where $\lfloor L \rfloor$ denotes the largest integer less than or equal to L .



Indian Institute of Technology,
Delhi

Ranjan Bose
Department of Electrical Engineering

So, this is the entropy function H of x minus $x \log$ to base 2 x minus $1 - x \log$ to the base 2 $1 - x$ right. And for a binary code with rate $1/n_0$, the minimum distance d_{min} satisfies this constraint $d_{\text{min}} \leq (n_0 v + n_0) / 2$. So, we basically get a feel that this n_0 has to be large, and the constraint length has to be large in order to have larger minimum distances, it gives us a feel for it.

(Refer Slide Time: 39:35)


Information Theory, Coding and Cryptography

Distance bounds

- An upper bound on d_{free} is given by (Heller, 1968)

$$d_{free} = \min_{j \geq 1} \left[\frac{n_0}{2} \frac{2^j}{2^j - 1} (v + j - 1) \right]$$

- To calculate the upper bound, the right hand side should be plotted for different integer values of j .
- The upper bound is the minimum of this plot.

 Indian Institute of Technology,
DelhiRanjan Bose
Department of Electrical Engineering


And there is a Heller bound for d_{free} , which is given as follows minimum j greater than or equal to 1 n_0 by 2 n_0 is the output frame, codeword frame 2 raised power j over 2 raised by j minus 1 n_0 plus j minus 1 n_0 is the constraint length. To calculate the upper bound, on the right hand side should be plotted for different integer values of j , so j is an integer. And the upper bound is the minimum of this plot.

(Refer Slide Time: 40:04)

Information Theory, Coding and Cryptography

Performance Bounds

- One of the useful performance criterion for convolutional codes is the bit error probability P_b .
- The bit error probability or the bit error rate (a misnomer!) is defined as the expected number of decoded information bit errors per information bit.
- Instead of obtaining an exact expression for P_b , typically, an upper bound on the error probability is calculated.
- We will first determine the **first event error probability**, which is the probability of error for sequences that merge with the all zero (correct) path for the first time at a given node in the trellis diagram.

 Indian Institute of Technology,
DelhiRanjan Bose
Department of Electrical Engineering

Now, let us look at some performance bounds. So, far we have looked at distance bounds. Now, we talked about performance bound in terms of the bit error probability P

b. So, we will first define, the first event error probability, which is the probability of an error for sequences that merge with all 0 correct paths for the first time at a given node in the trellis. So, that is called the first event error probability. So, we are trying to not find an exact expression for the bit error probability P_b , but an upper bound on the error probability.


(Refer Slide Time: 40:45)

Information Theory, Coding and Cryptography

Performance Bounds

- Let c' differ from the all zero path in d bits.
- Therefore, a wrong decision will be made by the maximum likely decoder if more than $\lfloor \frac{d}{2} \rfloor$ errors occur, where $\lfloor x \rfloor$ is the largest integer less than or equal to x .
- If the channel transition probability is p , then the probability of error can be upper bounded as follows.

$$P_d \leq \left[2 \sqrt{p(1-p)} \right]^d$$



Indian Institute of Technology,
Delhi

Ranjan Bose
Department of Electrical Engineering

Again this is again to give us a feel for the subject. So, let us c' differ from the all zero path in d bits. What is c' , it is the sequence received, what is transmitted, all zero path, without loss in generality, because we are looking at a standard linear code, which is convolutional code, and d errors have happened.

So, a wrong decision will be made by the maximum likely decoder, if more than $\lfloor \frac{d}{2} \rfloor$ errors occur, where x is the largest integer less than or equal to x . So, the channel transition probability is p , then the probability of error can be upper bounded as follows.

(Refer Slide Time: 41:35)


Information Theory, Coding and Cryptography

Performance Bounds

- Now, there would be many paths with different distances that merge with the correct path at a given time for the first time.
- The upper bound on the first error probability can be obtained by summing the error probabilities of all such possible paths:
$$P_e \leq \sum_{d=d_{free}}^{\infty} a_d P_d$$
where, a_d is the number of codewords of Hamming distance d from the all zero codeword.

Finally, we obtain

- $$P_e \leq T(D) \Big|_{D=2\sqrt{p(1-p)}}$$

 Indian Institute of Technology,
Delhi

Ranjan Bose
Department of Electrical Engineering

Now, there would be many paths with different distances that merge with the correct path at a given time. Please remember, there could be many paths that are in the trellis with different distances. And we have defined the first error probability as the 1, which merges back with the all zero path. So, we have an upper bound on the first error probability by summing all the error probabilities of such possible paths, so you have done that.

And if you remember, the number of paths, were easily found out by the modified state diagram, and $T(D)$ was expression we had found out, so we now, obtain the upper bound $P \leq T(D)$, where D is equal to $2\sqrt{p(1-p)}$. So, this is a very quick way, we already had found out how to calculate $T(D)$ modified state diagram, and then you can quickly get an estimate on the upper bound on the probability of error.

(Refer Slide Time: 42:31)

Information Theory, Coding and Cryptography

Performance Bounds

- The bit error probability, P_b , can now be determined as follows.
- P_b can be upper bounded by weighting each pairwise error probability, P_d , by the number of incorrectly decoded information bits for the corresponding incorrect path n_d .
- For a rate k/n encoder, the average P_b is


$$P_b \leq \frac{1}{k} \sum_{d=d_{free}}^{\infty} a_d n_d P_d$$

- It can be shown that

$$\left. \frac{\partial T(D, I)}{\partial I} \right|_{I=1} = \sum_{d=d_{free}}^{\infty} a_d n_d D^d$$

- Thus,

$$P_b \leq \frac{1}{k} \left. \frac{\partial T(D, I)}{\partial I} \right|_{I=1, D=2\sqrt{p(1-p)}}$$

 Indian Institute of Technology, Delhi
Ranjan Bose
Department of Electrical Engineering

So, what we can do is if you are talking about P_b , we upper bound P_b in terms of the pairwise error probability. And with a little bit of algebra you can look at that $\frac{\partial T}{\partial I}$, where this is the augmented modified state diagram $T(D, I)$ again evaluated I equal to 1 and D is equal to $2\sqrt{p(1-p)}$. So, that gives you an upper bound on the P_b .

(Refer Slide Time: 43:08)

Information Theory, Coding and Cryptography


Octal Representation

- The octal notation for the generators of the $R = 1/2$, $v = 4$ encoders are 15 and 17
- The **octal 15** can be deciphered as $15 = 1-5 = 1-101$.
 $g_1(D) = 1 + (1)D + (0)D^2 + (1)D^3 = 1 + D + D^3$.
- Similarly, $17 = 1-7 = 1-111$.
- Therefore,

$$g_2(D) = 1 + (1)D + (1)D^2 + (1)D^3 = 1 + D + D^2 + D^3$$

Thus,

$$G(D) = [1 + D + D^3 \quad 1 + D + D^2 + D^3]$$

 Indian Institute of Technology, Delhi
Ranjan Bose
Department of Electrical Engineering

Now, we change gears and we talk about an octal representation. And we will now look at some of the known good convolutional codes. So, in literature they are succinctly

written in terms of the octal representation. So, we show it by an example. It is just a matter of representing the generator polynomial matrix for convolutional code.

Suppose rate is half n is 4, and we say the octal notation for this convolutional code is 15 and 17, what does it mean. Well we are trying to say something, we are trying to tell you the components of the generator polynomial matrix. So, octal 15 is 15, 15 is nothing but 1 dash 5 that is nothing but 1 dash in binary 1 0 1. So, we have this first 1 as 1, and then 1 0 1 if you see is the coefficient for D , D squared and D cubed. So, this first 1 is the coefficient for D raise power 0, so you have 1 1 0 1 just look at the coefficients of D raise power 0 D raise power 1 D raise power 2 D raise power 3. So, the g_1 will be 1 D D cubed.

Similarly, 17 can be broken up and as 1 dash 7 is 1 dash 1 1 1, all coefficients are present, so it is 1 1 D raise power 1 1 times D raise power 2 1 times D raise power 3, so it is 1 plus D plus D squared plus D cubed, it is just a representation. So, this g_2 was written as 17, and g_1 was written as 15. So, if I say, the encoder is 15 and 17, I am just writing $G D$ as 1 plus D plus D cubed 1 plus D plus D square plus D cube, it is a rate 1 by 2, so you only need 2 entries for this generator polynomial matrix. So, it is a very simple octal representation for this $G D$. And this $G D$, we know very well how to write using a circuit.


(Refer Slide Time: 45:26)

Information Theory, Coding and Cryptography

Known Good Convolutional Codes

- Rate $\frac{1}{2}$ codes with maximum free distance.

	v	n	Generators (octal)		d_{free}	Heller Bound
Non-Catastrophic	3	6	5	7	5	5
	4	8	15	17	6	6
	5	10	23	35	7	8
	6	12	53	75	8	8
	7	14	133	171	10	10
Catastrophic	5	10	27	35	8	8
	12	24	5237	6731	16	16
	14	28	21645	37133	17	17

 Indian Institute of Technology, Delhi
 Ranjan Bose
Department of Electrical Engineering

So, here we show, known good convolutional codes. both catastrophic, and non-catastrophic codes with rate 1 by 2, and we write down the maximum free distance. We already understand how octal is written. This is the example, we did just now if 15 and 17 is the octane generator, then what it means for the rate 1 by 2 for that the d_{free} is given, and the subsequent Heller bound is given. So, we see, the Heller bound is pretty good for most of these cases.

So, what it means is that if I have to look at a rate 1 by 2 convolutional encoder, and if I have reasonable amount of compute power, because n_u , which is the constraint length directly has a implication on the decoding complexity. If n_u goes up, the states in the trellis goes up exponentially, and then I have to keep track of more number of nodes, while we do Viterbi decoding. So, we choose the constraint length based on our computational strength at the decoder. And if you see, at the cost of higher constraint length, we have a better d_{free} leading to more error correction.

(Refer Slide Time: 46:58)

Table 6.3: Rate 1/3 codes with maximum free distance. Information Theory, Coding and Cryptography

Known Good Convolutional Codes

- Rate 1/3 codes with maximum free distance.

	v	n	Generators (octal)			d_{free}	Heller Bound
Rate 1/3	3	9	5	7	7	8	8
	4	12	13	17	17	10	10
	5	15	25	37	37	12	12
	6	18	47	75	75	13	13
	7	21	133	175	175	15	15

Indian Institute of Technology, Delhi
 Ranjan Bose
Department of Electrical Engineering

Similarly, if you look at rate 1 by 3 encoder, we have again constraint lengths, and octal generator. But, clearly there should be three entries now, because it is rate 1 by 3, so there should be 3 G 1 D, G 2 D and G 3 D for the generators. And again you have got this d_{free} constraint here. So, rate goes down, and you have a higher d_{free} here. And these are the best in their class. So, they have been found by computer searches, and this is the best d_{free} that you can get with this.

(Refer Slide Time: 47:33)

Information Theory, Coding and Cryptography

Summary

- Matrix Description
- Viterbi Decoding Codes
- Bounds
- Known good convolutional Codes
- Examples

Indian Institute of Technology, Delhi 48 Ranjan Bose
Department of Electrical Engineering

So, now we summarize what we have done in today's class. We started off with the matrix description of convolutional codes. Then we looked at different decoding strategies, and we focused on the Viterbi decoding strategy for convolutional codes. Then we looked at bounds on minimum distance, and also performance bounds. Finally, we looked at some good known convolutional codes. We of course looked at some examples on the way.

So, with that we come to the end of this lecture.