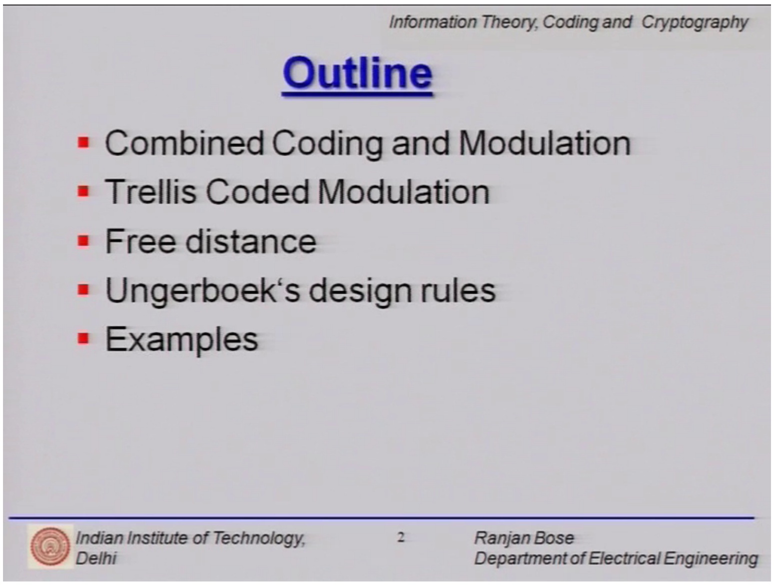
Information Theory, Coding and Cryptography
Dr. Ranjan Bose
Department of Electrical Engineering
Indian Institute of Technology, Delhi

Module - 30
Lecture - 30
Trellis Coded Modulation

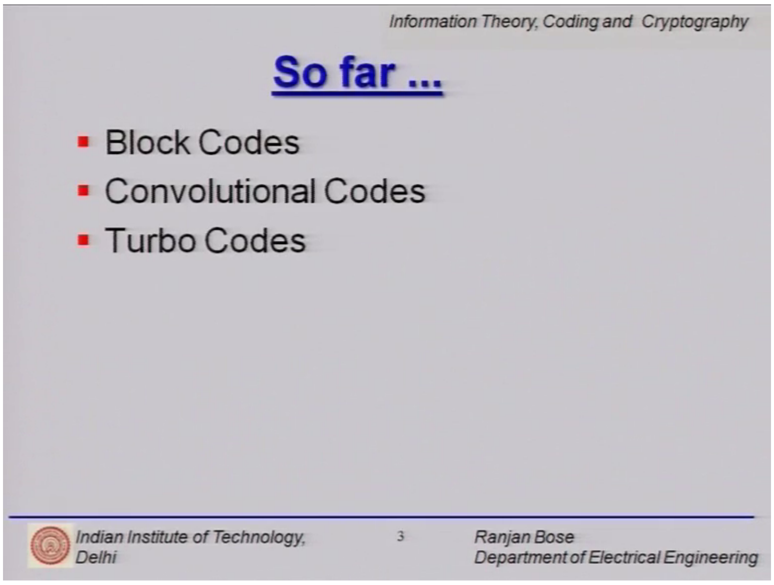
Hello and welcome to our lecture on Trellis Coded Modulation; let us start with the brief outline for today's talk.

(Refer Slide Time: 00:34)



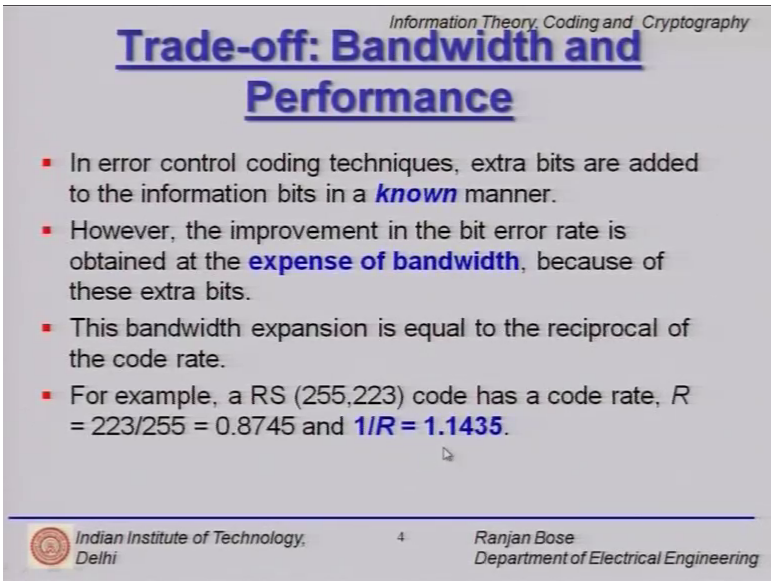
We would introduce this interesting notion of combining coding and modulation; so we will look at combined coding and modulation, then we will move on to the idea of trellis coded modulation, we will then characterize the free distance and come up with interesting idea proposed by Ungerboek in terms of his design rules; of course we look at some examples as we go along.

(Refer Slide Time: 01:03)



So, so far what have we done? We have looked at block codes, linear block codes, we look we have looked at cyclic codes. BCH codes, then we changed gears and looked at codes with memory and we looked at convolutional codes and turbo codes and now we move on to a completely different area where we look at coding and modulation combined together.

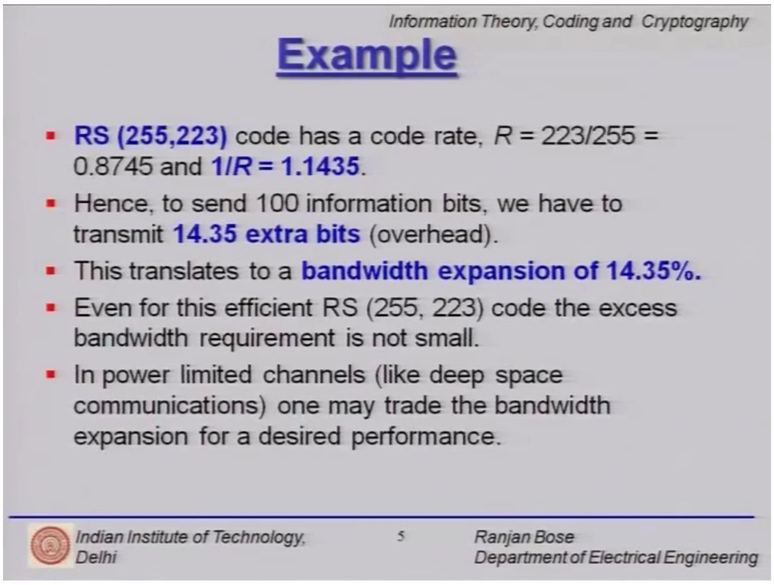
(Refer Slide Time: 01:30)



But why do we do that? So first let us look at the tradeoff between bandwidth and performance; we know that in all error control coding techniques extra bits are added to the information bits in a known manner therefore, the receiver can decode it.

But this is not free additional bits come at the expense of additional bandwidth and as we know bandwidth is not cheap, the bandwidth expansion is equal to the reciprocal of the code rate for example, if you have the Reed Solomon 255 comma 223 code then there is a 1 over R equal to 1.14 so as a 14 percent expansion.

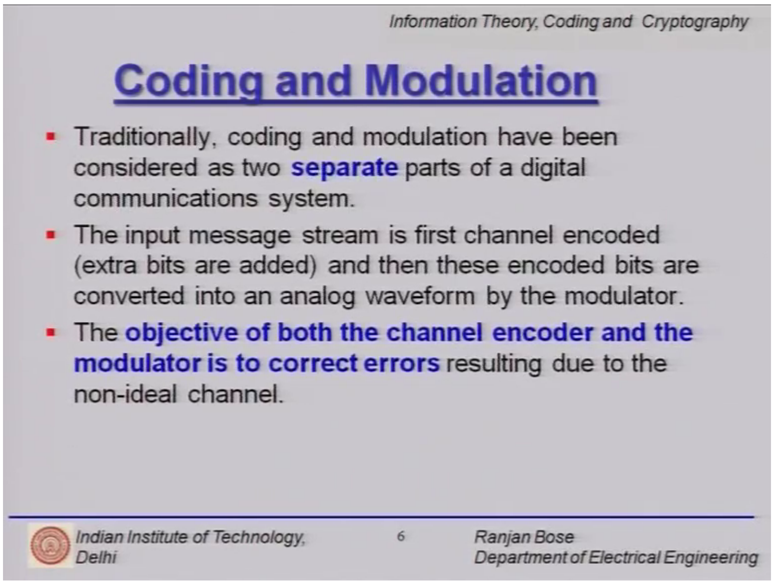
(Refer Slide Time: 02:14)



Now, this tantamounts to for a every 100 information bits we have about 14.35 overhead bits on an average.

So, a 14.35 percent expansion of bandwidth this is non-trivial even for an efficient code like Reed Solomon 255 comma 223. So in power limited channel one may trade the bandwidth expansion for the desired performance, but for bandwidth limited channels we may not have this luxury. So can we do better for example, telephone lines are bandwidth limited and if we want to push data over telephone lines we need to work more smartly.

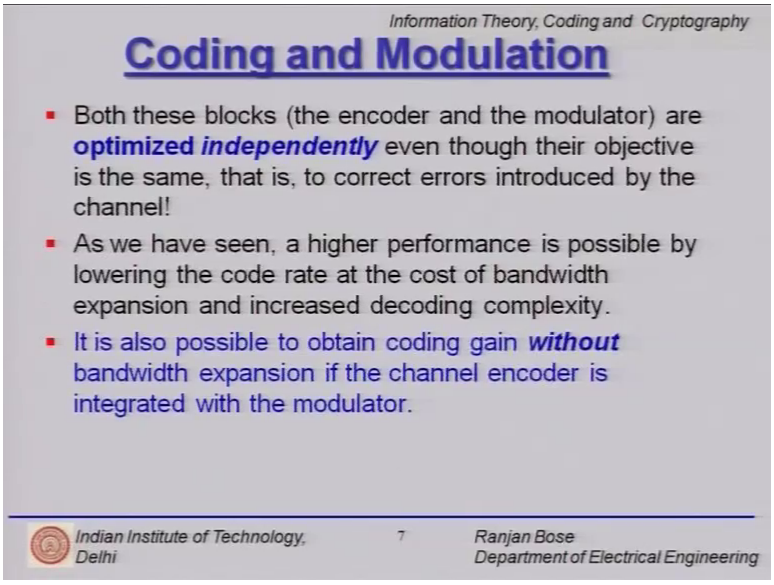
(Refer Slide Time: 03:01)



So, let us look at coding and modulation; how can we talk about coding and modulation on the same slides. Well traditionally coding and modulations have been considered as two separate parts of a digital communication system; you first do coding and then follow it up by modulation right. First the input stream is channel coded and then we convert into an analog waveform by the modulator, but if you look at it more carefully the objective of both the channel encoder and the modulation is to correct errors and this is because of the imperfections in the channel, so both of them are doing the same job.

If you look at an error control code, we find out what is the probability of residual error; if we look at the characteristic of a modulator we always plot the BER versus the SNR curve; the standard waterfall curve, so again we are talking about bit error rate.

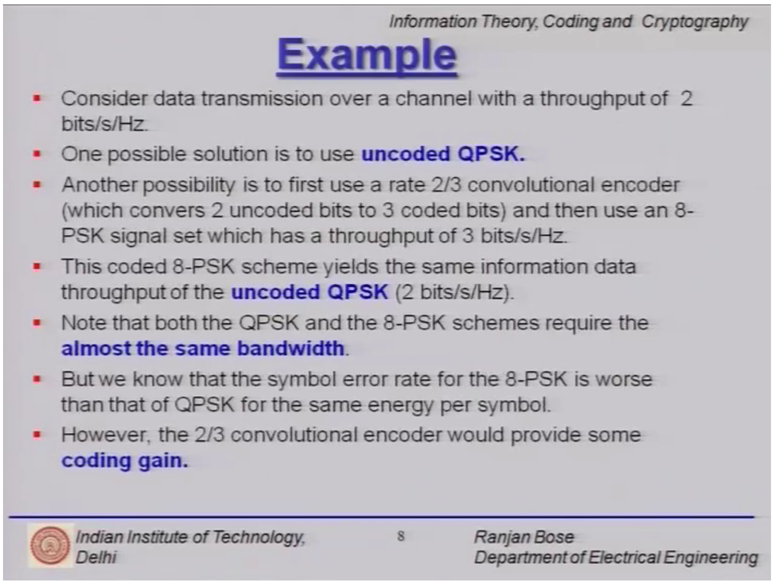
(Refer Slide Time: 04:05)



So, now both this blocks are optimized independently even though it seems their objective is the same. Now a high performance is possible by lowering the code rate at the cost of the bandwidth expansion and increase decoding complexity that is what we have learned so far.

So, if you need a stronger code, you have to have a poorer code rate because that is how you will be able to recover from more number of errors. Now the question is; is it possible to obtain this coding gain without the bandwidth expansion, is there a free lunch, can we get something out of nothing; that is the question we are asking whether both coding and modulation can be looked at together to give you this gain without additional bandwidths requirement.

(Refer Slide Time: 05:01)

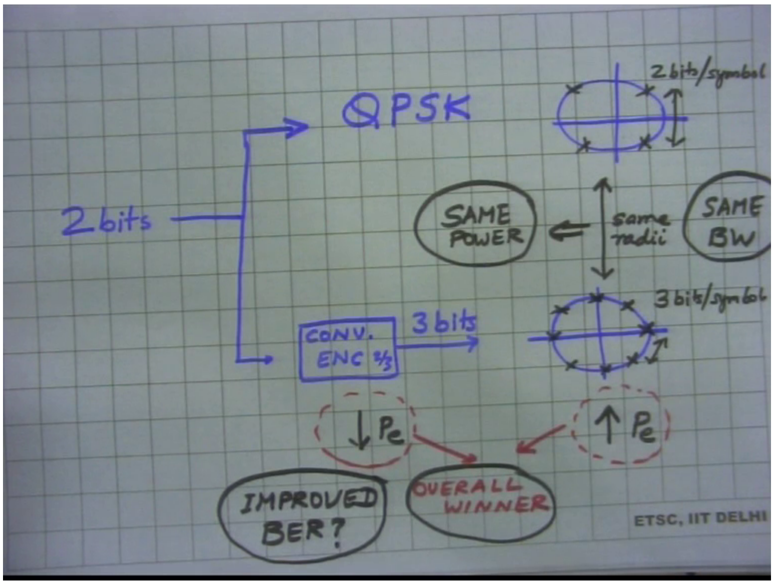


So, let us understand this with a simple example; suppose we are looking at data transmission over a channel of throughput 2 bits per second per hertz ok. So the moment I see 2 bits per second per hertz my mind jumps to uncoded QPSK; where we have 2 bits per symbol, but we say wait a minute we can do this differently we first use a 2 by 3 rate convolution encoder and it takes the 2 bits of raw data and converts into 3 bits of coded data, and then 3 bits have to be transported and we use now an 8-PSK because I need 3 bits per symbol because I do not need additional bandwidth.

So, this coded 8-PSK schemes yield the same information data throughput as the uncoded QPSK which was sending 2 bits per second per Hertz. We assume that QPSK and 8-PSK are consuming roughly the same bandwidth it is not a bad assumption, but we know that the symbol error rate for 8-PSK is worse off than that of QPSK for the same energy per symbol, but on the other hand the 2 by 3 convolution encoder has been able to provide some coding gain.

So, if you look at it graphically, this is what we did, we had this 2 bits coming in now if you look at these 2 bits these 2 bits can be either sent using QPSK where you know there is a constellation diagram and we have these 4 possible symbols in the constellation diagram and you have got 2 bits per symbol.

(Refer Slide Time: 06:54)



But we say no wait a minute this 2 bits can be passed through a convolution encoder which is rate 2 by 3, so 2 bits go in and 3 bits come out.

So, what do I do I have to modulate and send so I need a a higher order modulation scheme so I need probably an 8-PSK so if you look at it this is my 8-PSK this is the constellation diagram. Here I have got 3 bits per symbol please note that this radius of the circle is proportional to the energy per symbol; so we do not want additional power so the radii are the same; so the same radii implies that this circle has these symbols closer together as opposed to the distance.

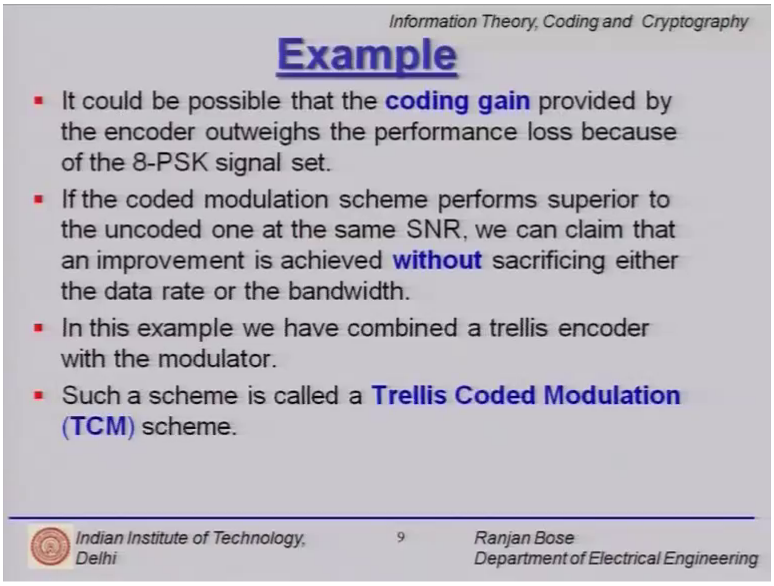
So, the Euclidean distance goes down, the Euclidean distance going down may lead to an increase in probability of error, but you already have an error control code put into place whose job is to reduce the error, so this block is going to reduce the probability of error. So on one side we have a mechanism that tries to push down the error the other side may lead to increase of error.

So, if you play our cards right then we can come out an overall winner; so this trade off will decide whether we finally, end up having a better BER. Please note we have not touched power, so same power and here these two modulation schemes we have not really asked for additional bandwidth so we have same bandwidth. Now with the same power and same bandwidth do we have an improved BER that is the question; did we really gain something out of nothing.

We know that from Shannon's theorem you give me more power I will give you improved BER, you give me more bandwidth I will give you an improved BER, but I ask for no additional power, no additional bandwidth and am I going to get an improved BER; so the answer is yes and let us see how we can do it. So we go back to our slide and what we have just mentioned it could be possible that the coding gain provided by the encoder outweighs the performance loss because of the 8-PSK signal set as we are looking in our slides.

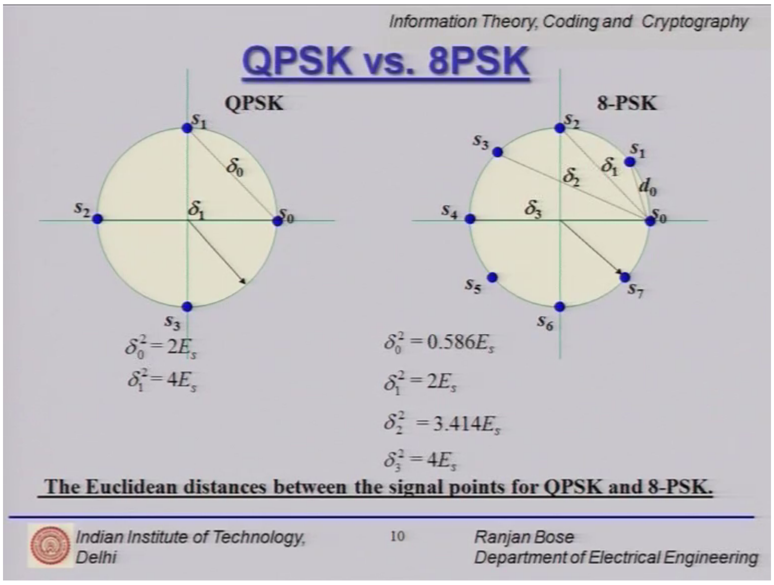
Now if the coded modulation scheme performs superior to the uncoded one at the same SNR we can claim that an improvement is achieved without sacrificing either data rate or bandwidth.

(Refer Slide Time: 11:05)



So please note we did not sacrifice data rate in either case we got those 2 bits per second through the channel. So in this example we have combined a Trellis encoder with a modulator and hence we have looked at combined coding and modulation; this scheme is called a Trellis Coded Modulation or TCM for short.

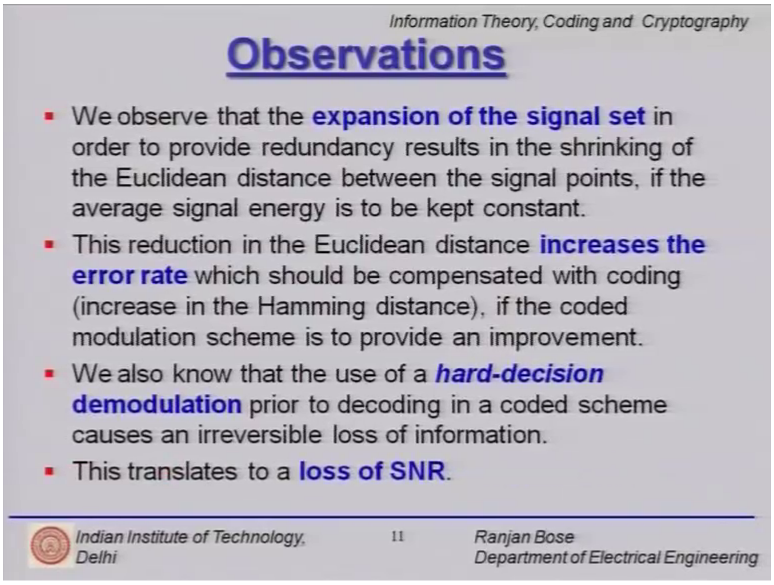
(Refer Slide Time: 11:48)



So, let us look at QPSK and 8-PSK a little bit more carefully; if you see that both the circles are of the same radius which means that energy per symbol is the same. Now we also know that the probability of error depends on the Euclidean distance, here the smallest Euclidean distance between s 1 and s 0 will decide what is the error rate. So I put delta 0 as the first distance if the radius of the circle is under root E s then the distance between s 1 and s 0 delta naught squared would be 2 E s and similarly the second possible distance between s 2 and s 0 will be 4 E s.

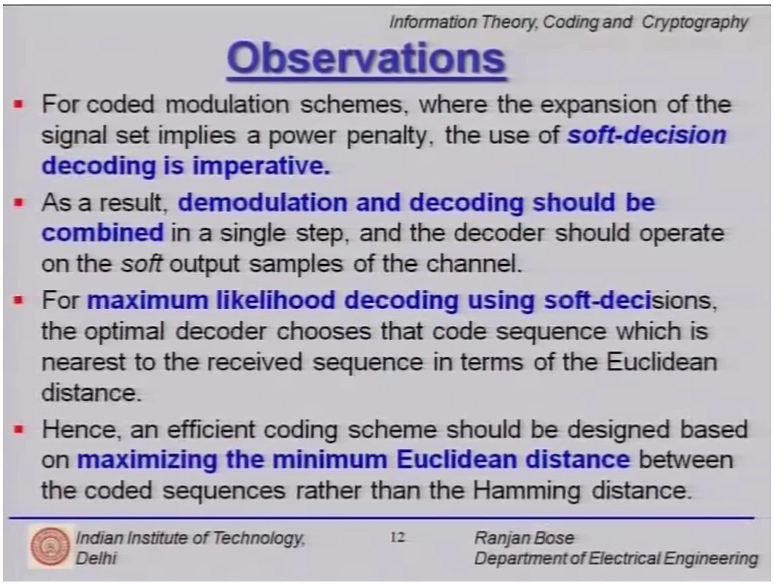
On the other hand if you look at 8-PSK the distances are shorter, so the smallest distance possible delta naught 0 is 0.586 E s, where E s represents the energy per symbol and the radius of the circle is under root E of s. Here delta 1 squared is between s 0 and s 2 same as delta 0 squared in QPSK, but we have 2 other distances delta 2 and delta 3 listed here. So, this smaller distance leads to a higher error for 8-PSK.

(Refer Slide Time: 13:15)



So, we make some observations, we observe that the expansion of the signal set in order to provide redundancy results in the shrinking of the Euclidean distance; this leads to an increase in the error rate which must be compensated with the coding gain; that is the increase in the Hamming distance. We also know that the note that the use of hard decision demodulation prior to decoding in a coded scheme causes an irreversible loss of information. So, these are the observations we make and finally, this hard decision decoding leads to loss of SNR.

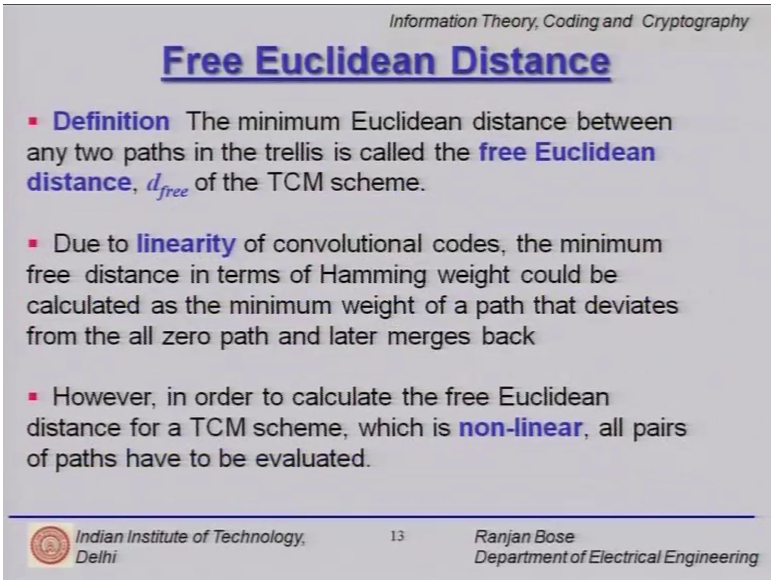
(Refer Slide Time: 13:51)



Some more observations so for coded modulation schemes where the expansion of the signal set implies power penalty, use of soft decision decoding is imperative ok; as a result demodulation and decoding should be combined. So not only the modulation and coding is combined the demodulation and decoding itself is combined as the receiver side. For the maximum likelihood decoding using soft decision the optimal decoder chooses that sequence which is nearest to the received sequence this time in terms of the Euclidean distance and not the Hamming distance that we saw in the last time.

So, what is an efficient coding scheme? It should be based on maximizing the minimum Euclidian distance between coded sequence rather than the Hamming distance.

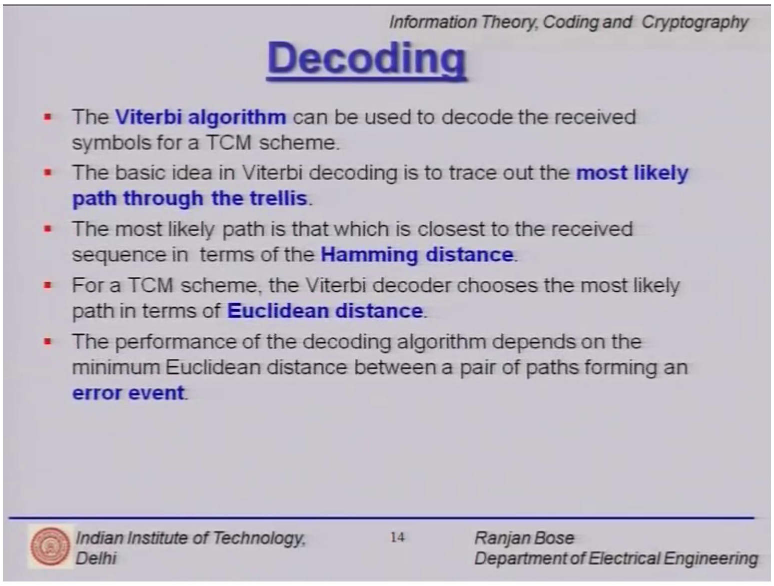
(Refer Slide Time: 14:42)



We have looked at free distance earlier in terms of convolution codes we define the free Euclidean distance; so the minimum Euclidian distance between any 2 paths in the trellis is called the free Euclidean distance d free of the TCM scheme. Please note here since there is a convolution encoder present they will be a trellis diagram, but the labels of the branches instead of being bits should be symbols because, all the bits have been mapped to symbols.

Now, you must remember in convolution codes we had the linearity constrained; so we were working with linear codes we could take the all 0 paths and find out the d free with respect to the all 0 path. Here is a word of caution for TCM; TCM is non-linear therefore, in order to find the d free one must look at all possible pairs within the trellis diagram ok. We are looking at non-linear codes this mapping mapping of the bits to the symbols introduces the non-linearity.

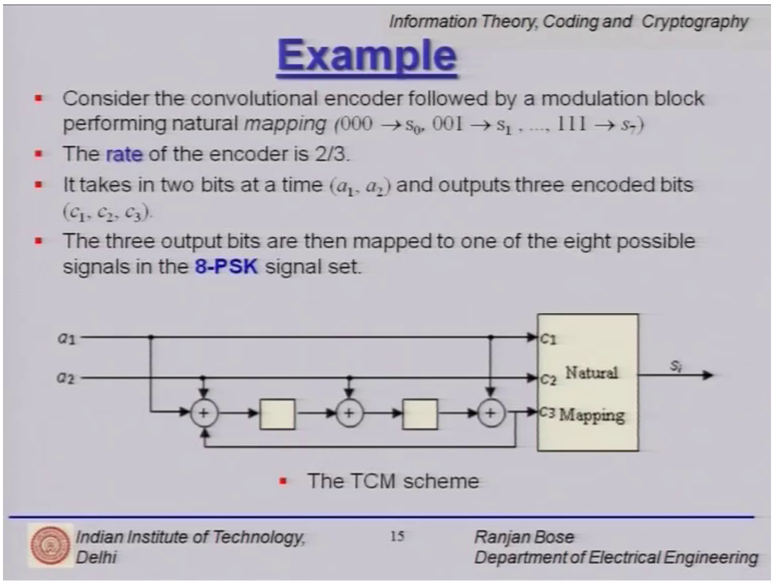
(Refer Slide Time: 16:03).



So let us look at the decoding; please note there is a viterbi algorithm associated with it simply because we have a trellis diagram which is used for encoding. So we do the decoding using the viterbi algorithm which we have studied already in the context of convolutional codes. What does the viterbi decoding do? It looks for the most likely path through the trellis, but this time instead of finding out the most likely path closest to the received sequence in terms of Hamming distance we should be using the Euclidean distance for TCM.

The performance of the decoding algorithm depends on the minimum Euclidian distance between the pairs of paths forming an error event. What is an error event? First the paths diverge and then they merge back ok.

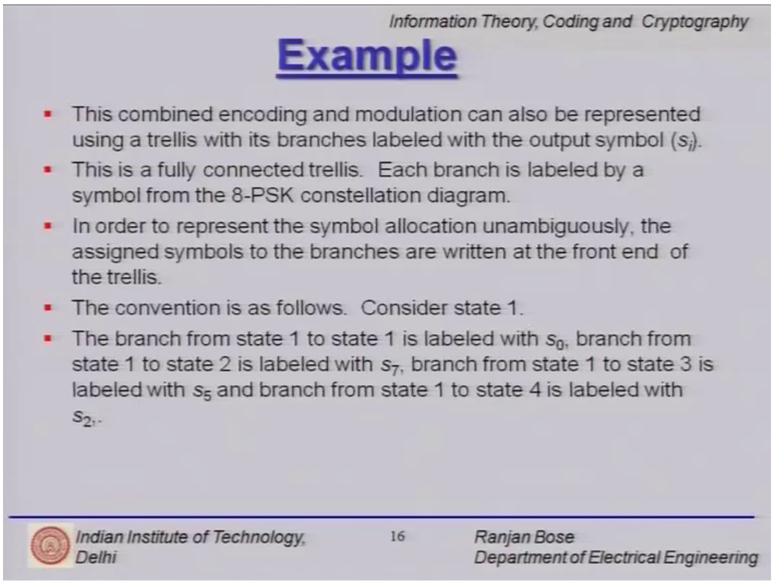
(Refer Slide Time: 16:50)



So we come back to our slide; we have done all of so let us look at this example now, let us consider a simple example where if you see a trellis encoder in the beginning it takes in 2 bits a 1 and a 2 and gives out 3 coded bits c 1, c 2, c 3.

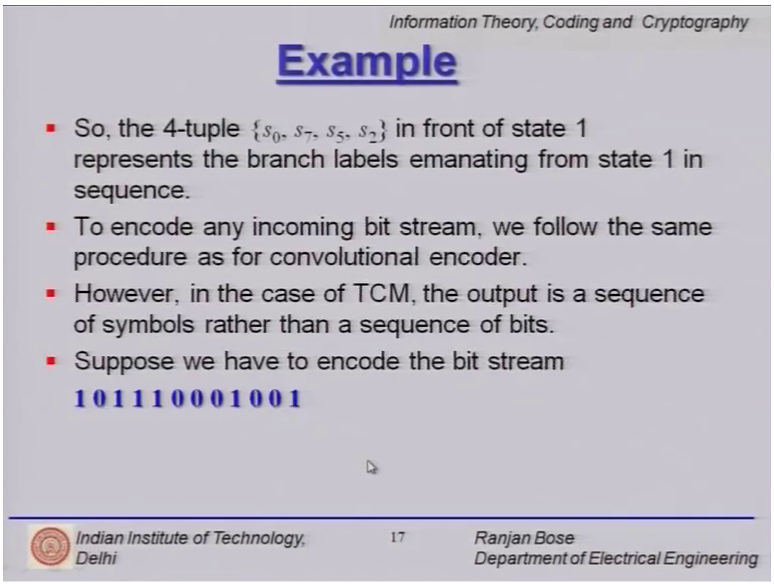
But the moment I get these 3 bits I use a natural mapping on 8-PSK and it gives me a symbol. So if you look at the combined coding and modulation 2 bits lead to 1 symbol; what is natural mapping? It is 0 0 0 is s 0, 0 0 1 bit means transmit s 1, these are the symbols 0 1 0, transmit s 2 up to 1 1 1 means transmit s 7 symbol from the constellation diagram. The rate is clearly 2 by 3 and it takes in 2 bits and converts it into 3 bits and the constellation is 8-PSK.

(Refer Slide Time: 18:17)



So, if you see that this combined encoding and modulation can be represented using a trellis diagram, but this time we label the branches with the output symbols and this is a fully connected trellis where each branch is labeled by the symbols from 8-PSK and encoding is a simple process as we had learnt earlier; 1 comes in you take the upper branch, 0 comes in take the lower branch and you read out what is written on top of the branch.

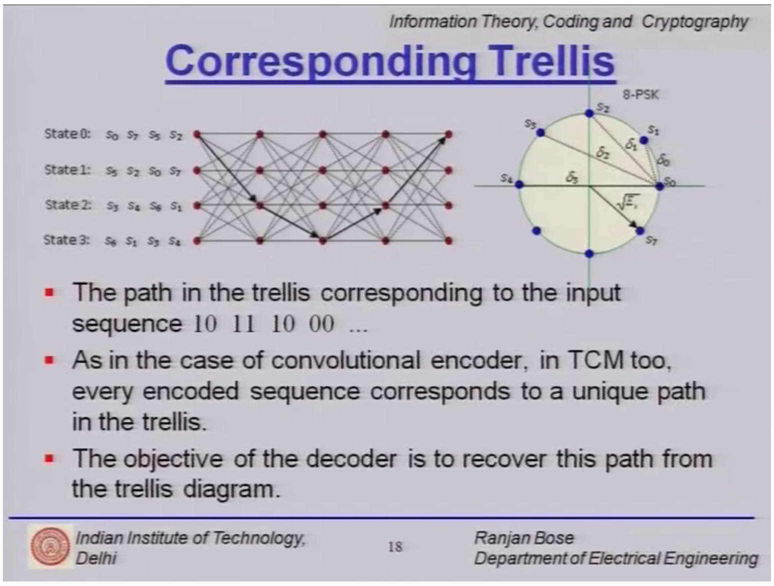
(Refer Slide Time: 18:53)



So, if you have to look at the trellis diagram you just take in the input bit stream which tells you which path in the trellis you have to take.

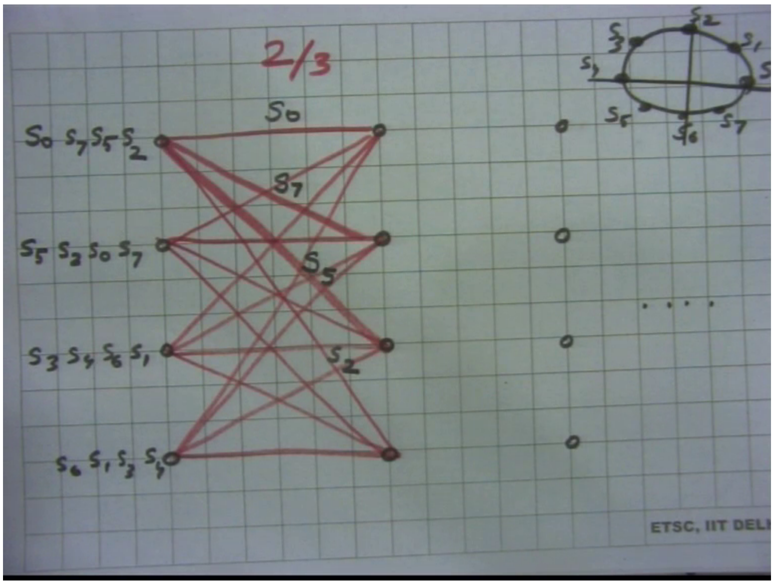
Let us take a simple example of a streak sequence of bits to be encoded using the trellis coded modulation scheme, say the bit stream is 1 0 1 1 1 0 0 0 1 0 0 1 this has to be encoded using the trellis coded modulation scheme.

(Refer Slide Time: 19:25)



So, here is the corresponding trellis for the TCM schemes; now please note that instead of labeling each branch and it becomes a little confusing we write the symbols here so s 0 corresponds to the first branch label, s 7 corresponds to the second branch, s 5 third and s 2 forth. So, let us look at it graphically here so we draw this trellis diagram.

(Refer Slide Time: 20:01)



There are 4 states in the trellis, because if you see you have 2 memory units leading to 4 states and it is a fully connected trellis which means that each node is connected to every other node and so and so forth. Now for any bit that comes in and this is a rate 2 by 3; so 2 bits come out so they have possible 4 possible transitions 0 0 comes in take this branch, 0 1 take this branch, 1 0 take this branch, 1 1 take the last branch.

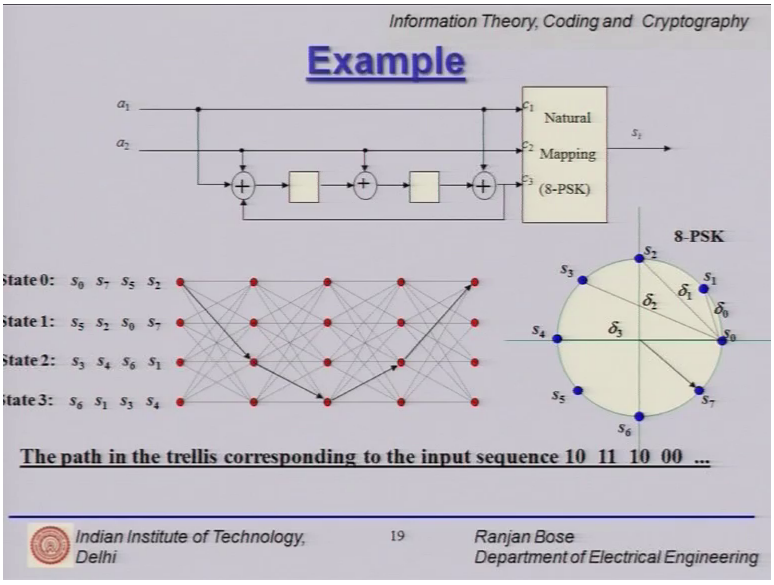
What happens once you take the branch you get an output bit, but you encoded using and then you map it back using the 8-PSK natural mapping. So the mapping strategy that we have given is s 0 for 1 what is this s 0 well if you look at your 8-PSK these are your symbols; so this is your s 0 this is your s 1 so this s 0 is this s 0, but suppose you go for this transition then you label it with s 7. This is a design that we have come up with I could have labeled it something else, but we are not yet looking at the design criteria and then the third branch is s 5, the fourth branch is s 2.

If I start labeling all of them it gets a little congested so we can say this is s 0 s 7 s 5 and s 2 so this is the convention. Similarly for the next 1 I can write s 5 s 2 s 0 s 7 of course, it will be easy to start connecting them and labeling and so and so forth. And similarly the third one and the fourth one and this repeats, so we just fix it and write out the trellis s 4 s 6 s 1 similarly s 6 s 1 s 3 and s 4 right.

So, we can always write a trellis diagram in a compact form; so let us look at the slides again and if you see this fully connected trellis then we must now find out a way to encode the path in the trellis. So it says the bits input bit stream is 1 0, 1 1, 1 0 and 0 0 in the slide. So what we do is look at the first 2 bits since encoder and decoder are friends we know that 2 bits are encoded at one time and the encoder then takes 1 0 means takes the third path then 1 1 take this path and then so this basically gives which is the direction to take which sequence of bits will define that a certain path in the trellis must be traversed.

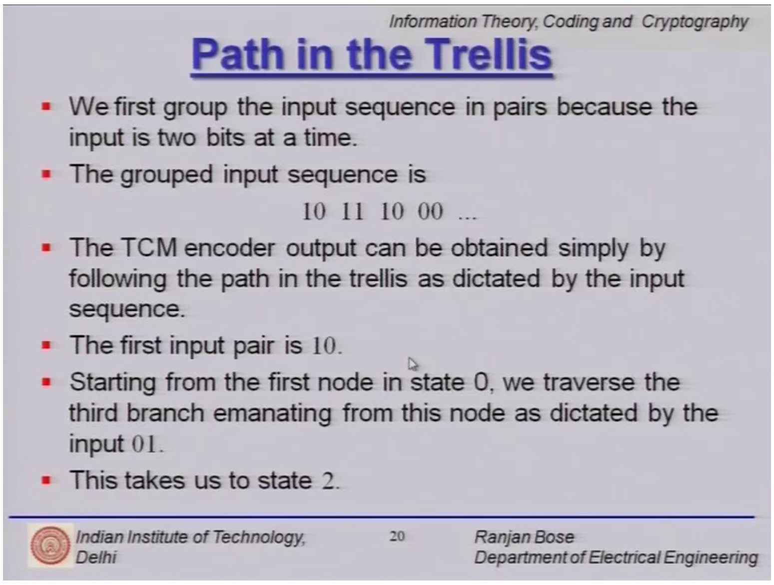
And what we do is simply read out what is written off on top of each of the branches. The objective of the decoder again is to recover this particular path from the trellis diagram.

(Refer Slide Time: 23:50)



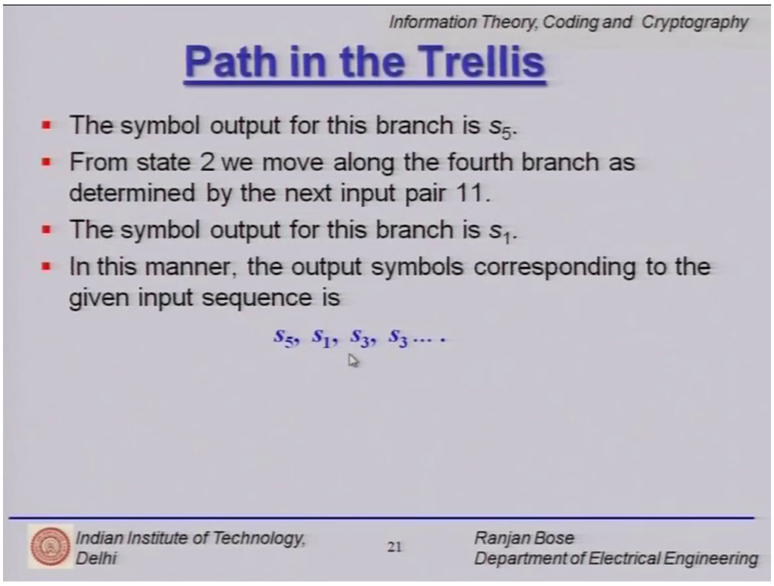
So, here is in a nutshell all of it if you see this is the hardware followed by the mapping, so I can describe my TCM scheme as a diagram with a circuit implementation and a mapping or equivalently I can define my trellis using the states and a trellis diagram they are the same you give me 1 and I will give you the other.

(Refer Slide Time: 24:11)



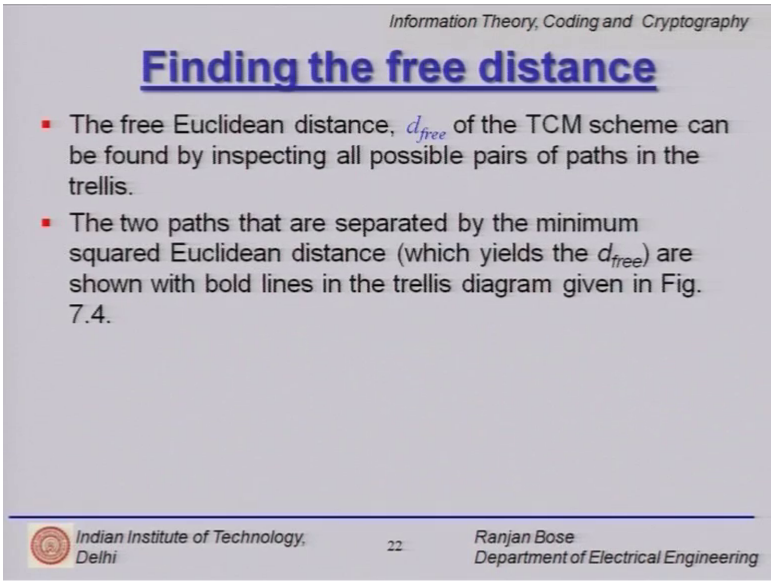
So, if you look at the path in the trellis this slide tells me how to go from the input sequence to the path on the trellis and the encoder takes 2 bits at a time and follows the consequent paths in the trellis.

(Refer Slide Time: 24:47)



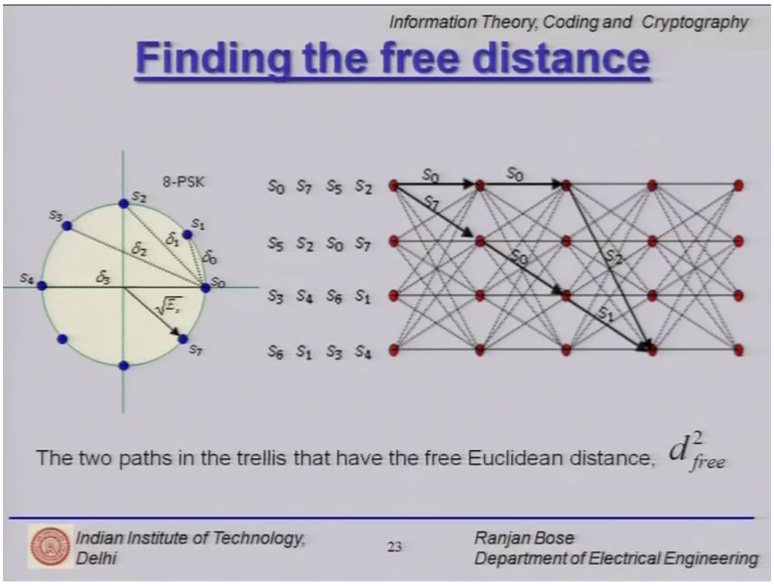
And reads out the corresponding symbol written on top of that branch so this slide basically tells you that first branch has s 5 written on it so read out s 5 second branch has written s 1 on it so read out s 1 s 3 and s 3 and so on and so forth.

(Refer Slide Time: 25:05)



So, now we go to the job of finding the free distance; we know that free distance is the single most important parameter which describes how good this trellis code is. So the free Euclidean distance of the TCM scheme can be found by inspecting all possible pairs of path in the trellis not just from the all 0 path this is because we are dealing with a non-linear code.

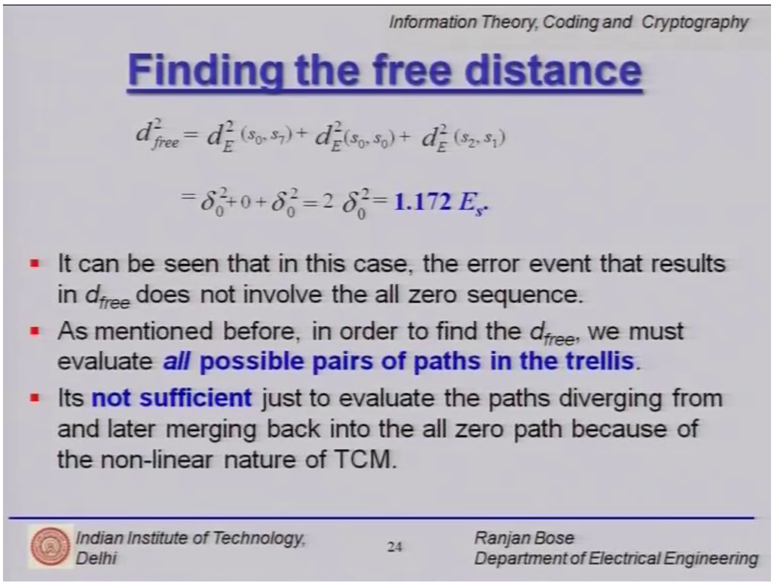
(Refer Slide Time: 25:39)



So let us look at this diagram and this is the same trellis coded modulation scheme described earlier here after a lot of inspection we have been able to identify 1 pair of path which leads to the minimum distance called the free distance.

So we have s 0 s 0 s 2 and s 7 s 0 s 1 that they have been shown using bold lines and clearly the distance is not coming from the all 0 path. In fact, the all 0 path does not lead to the calculation of d free here at all.

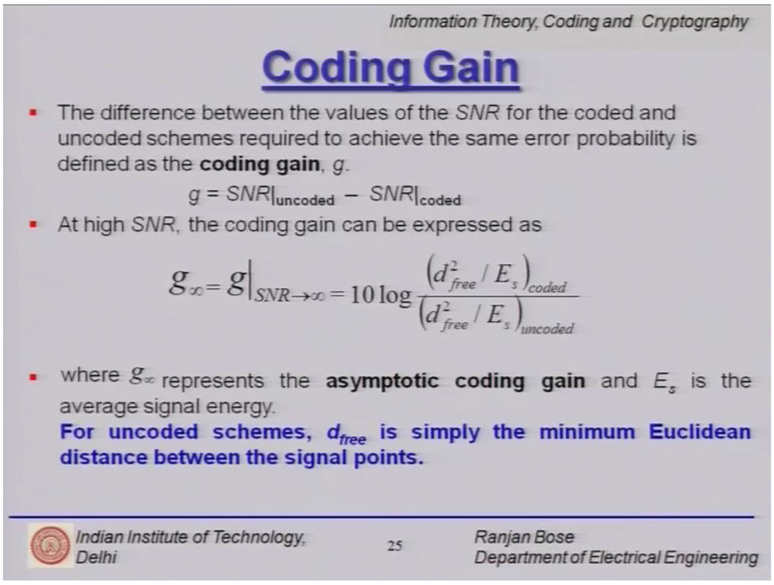
(Refer Slide Time: 26:20)



So, if you add those free distances, so it is Euclidean distance so if you see first we will take the distance between s 0 and s 7 plus between s 0 and s 0 plus distance between s 2 and s 1, but what distances are we adding? We are adding the Euclidean distances and therefore, we take the squared Euclidean distance because we can add the squared Euclidean distance overall.

So, we take the distance between s 0 and s 7 square Euclidean distance, square Euclidean distance between s 0 and s 0 and squared Euclidean distance between s 2 and s 1. If you add them up you get answer equal to 1.172 E s; what is E s? Well the radius of the circle is under root E of s so it can be seen that in this case the error event that results in the d free does not involve the all 0 path in the trellis and we looked at all possible pairs in the trellis to find out d free not just the all 0 path.

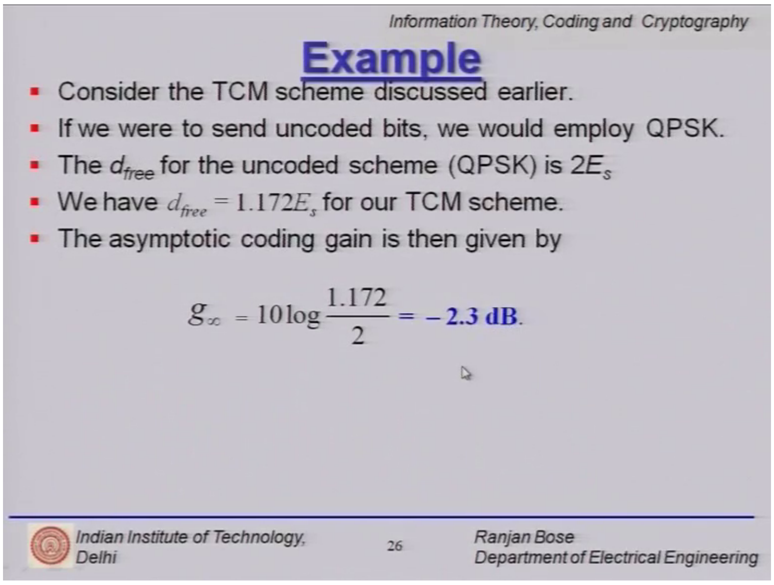
(Refer Slide Time: 27:30)



Now, we define something called as coding gain the difference between the values of the SNR for the coded and the uncoded schemes required to achieve the same error probability is called the Coding Gain. What is this gain? This g is SNR uncoded minus SNR coded we had seen this definition earlier also. So at high SNR this asymptotic coding gain g infinity where SNR tends to infinity is defined as 10 log to the base 10 d free squared over E s of coded scheme versus d free squared over E s of uncoded scheme this is called the Asymptotic Coding Gain.

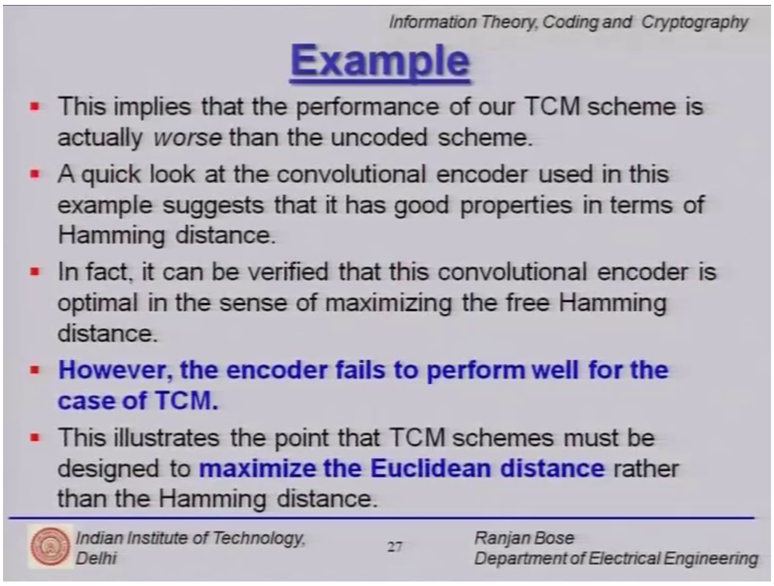
So, for uncoded scheme d free simply the minimum Euclidian distance between the signal points, so uncoded is in the denominator.

(Refer Slide Time: 28:28)



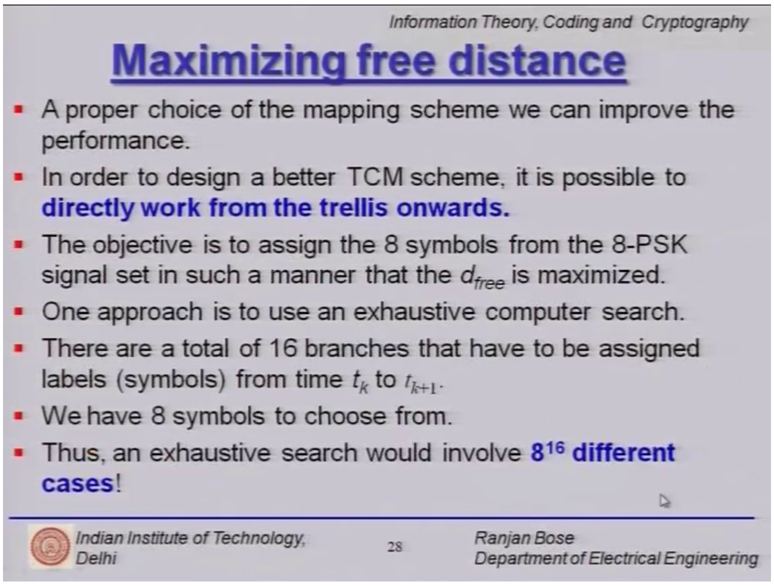
So, let us look at the TCM scheme we considered earlier, we had calculated d free as 1.172 E s for our scheme, but for QPS the distance between any 2 signal points is 2 E s; so g infinity is 10 log 1.17 divided by 2 equal to minus 2.3 dB; which means that we have actually lost out the gain provided by the encoder function encoder did not more than enough compensate the loss because of movement to 8-PSK. So, we did not kind of win the game we can do better.

(Refer Slide Time: 29:17)



And that is what we have to look at so our performance is actually worse than the uncoded scheme, so it is a 2 edged sword and it tells us that we have not we have used the good properties of the encoder in terms of Hamming distance, but not the Euclidean distance. So the aim of the design of good TCM scheme is to maximize the Euclidean distance; so we did not do well in terms of TCM; so we must now look at maximizing the Euclidean distance.

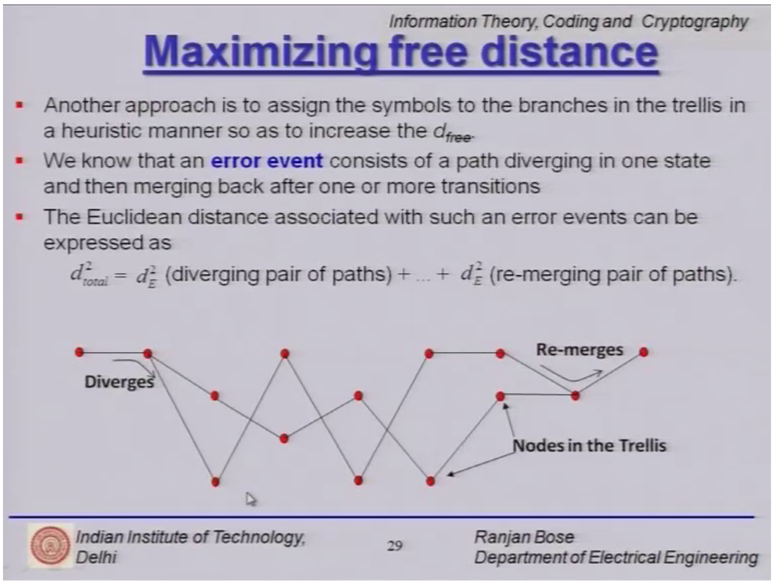
(Refer Slide Time: 29:49)



So how do we do that; well a proper choice of mapping scheme can improve the performance and it is better to design the TCM scheme working from the trellis itself because finally, it is a d free that we have to maximize and d free is calculated using the trellis so why not work with that trellis itself. So the objective is to assign 8 symbols of the 8-PSK signal set in such a manner that d free is maximized ok. Now in the last example there were 16 branches because there were 4 states 4 branches were coming out from each.

So, there were 16 branches we have to choose 8 symbols; so how many ways we can do it for an exhaustive search well 8 raise power 16 different cases that is just too large even a supercomputer, network supercomputer, distributed computing cannot solve that so we need better heuristic methods to solve it. A computer search will not work why are we making this point; for convolution encoders efficient computer searches had helped solve the problem, but not for this case.

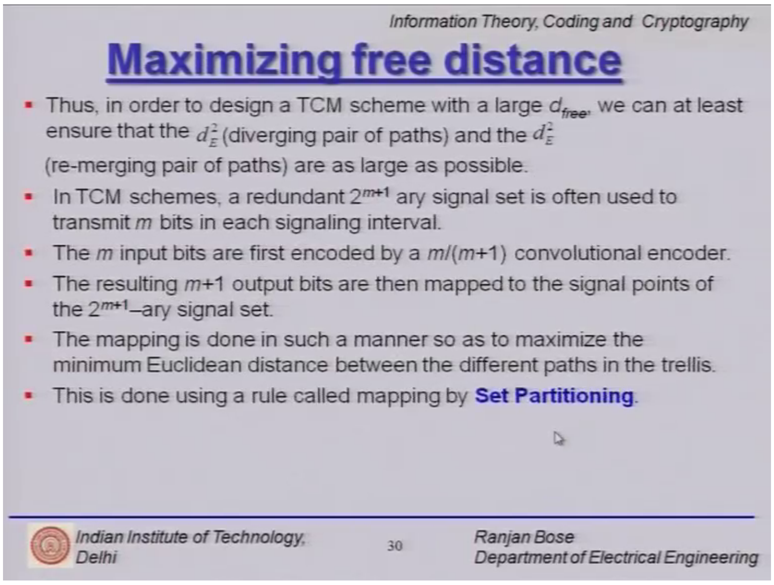
(Refer Slide Time: 31:01)



So, let us look at what is an error event will error event we have seen when 2 paths diverge they keep away from each other and then merge back in the trellis after sometimes. This red dots are the nodes in the trellis, so we have just shown 2 paths which diverge and merge path this is constitutes an error event and the one with the minimum distance between these 2 paths distance in terms of Euclidean distance is comes out as the d free. So at least what we can do we can maximize the two branches where it diverges and merges back and rest all will add to it, but we do not have much control how it jumps up and down in the trellis.

So, we will choose the Euclidean distance associated with at least the diverging branches and the merge in back branches to give us something and we also know that there will not be too many hop in 2 3 hops it combines back.

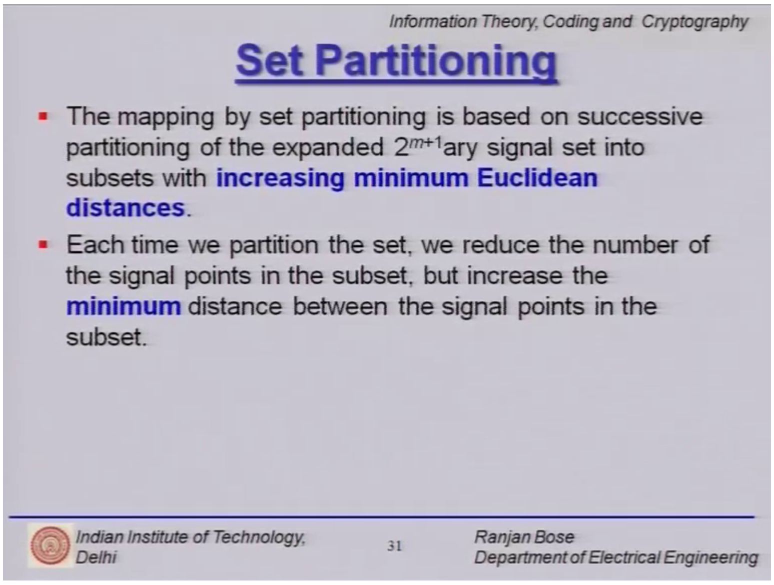
(Refer Slide Time: 32:07)



So, it will not be a bad choice so we look at the diverging d E diverging path and reemerging path and we have a redundancy of 2 raise power m plus 1 m-ary signal set is used to transmit m bits; why because we had this 8-PSK and then we went to from QPSK we went to 8-PSK.

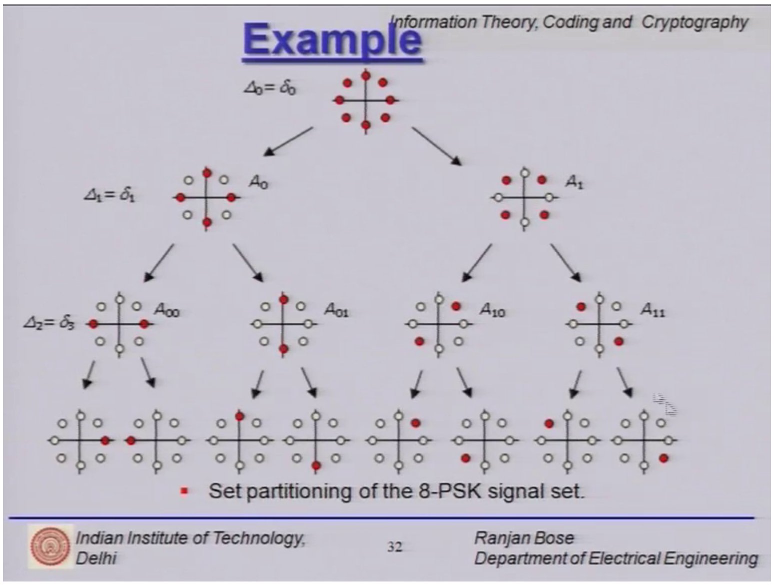
So, we have m over m plus 1 convolutional encoder in the last case we had 2 by 3 so m was 2 so m shows that uncoded and m plus 1 is the coded so this is a typical case you can have a 4 by 5, 7 by 8 convolution encoder. So the resulting m plus 1 output bits are then mapped so we will use this concept to come up with some design methodology in terms of increasing the Euclidean distance, but the method to do so is called set partitioning and let us discuss how it is done.

(Refer Slide Time: 33:11)



So, the aim of set partitioning is to progressively increase the Euclidean distance and then use that partitioned set to map onto the trellis; each time we partition the set we reduce the number of signals points, but increase the distance between the signal points in the subset.

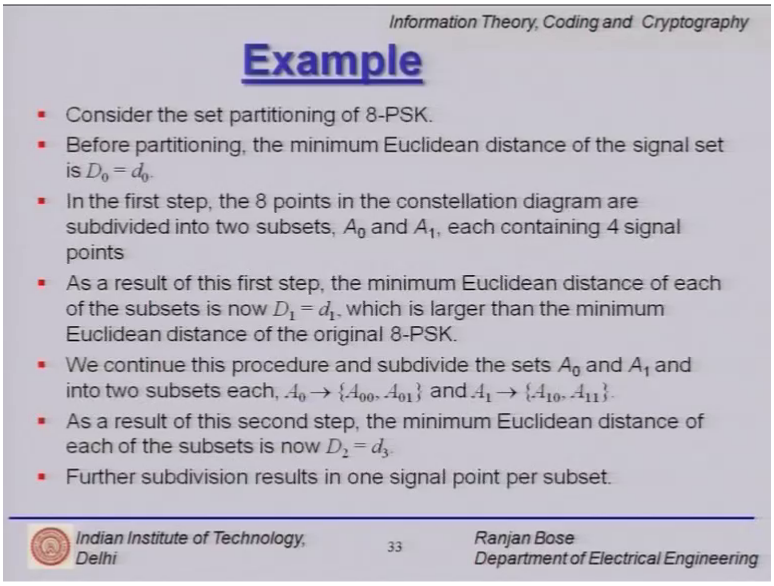
(Refer Slide Time: 33:33)



Let us look at an example let us start with 8-PSK on the top you can see 8-PSK they are all placed on a circle. So we now partition this is a set of 8 signal points which partition it into 2 4 and 4, but see the distance has increased so the minimum distance has gone up, but why should we stop here we can partition it further.

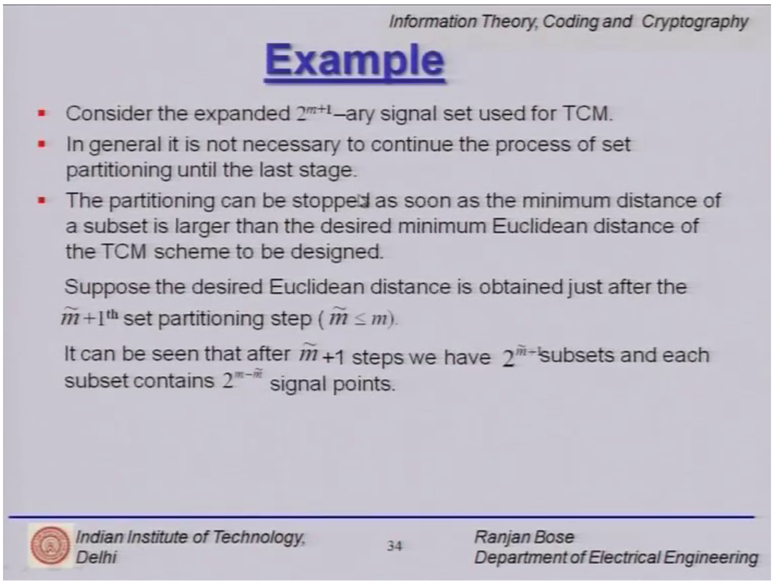
Again there is more than 1 way to do the partitioning, but we partition it in such a way that the minimum distance in this partition set between 2 constellation points have gone up so here there were 8 points now we have 4 points and 4 points highlighted by red and then 2 points and 2 points 2 points and 2 points and then of course, you have single point in the in the sets. So we have done the set partitioning of 8-PSK signal set it is pretty mechanical you can do it for any large constellation diagram.

(Refer Slide Time: 34:45)



So, now the aim is how to use this partitioned set and map it to the trellis that we have here; so what we do is we basically come up with a rule as to how to apply these points from the different levels of constellation diagram because level decides what is the minimum distance in terms of the diverging and merging paths.

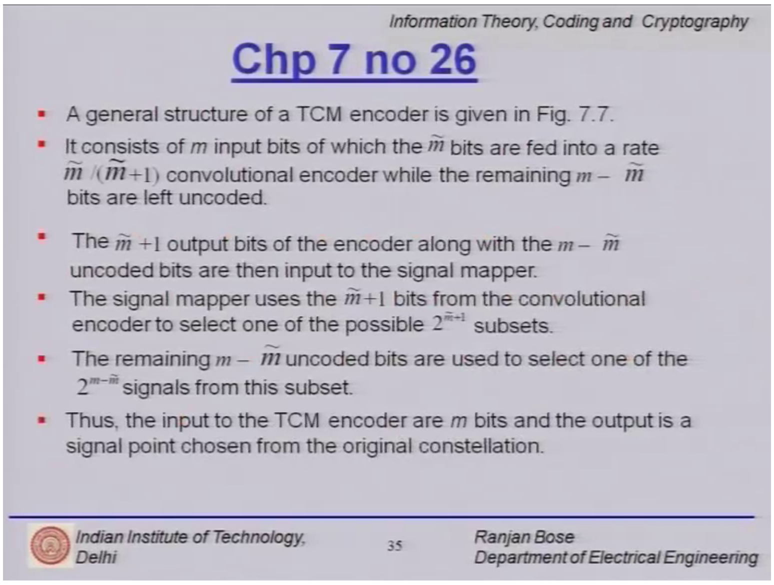
(Refer Slide Time: 35:16)



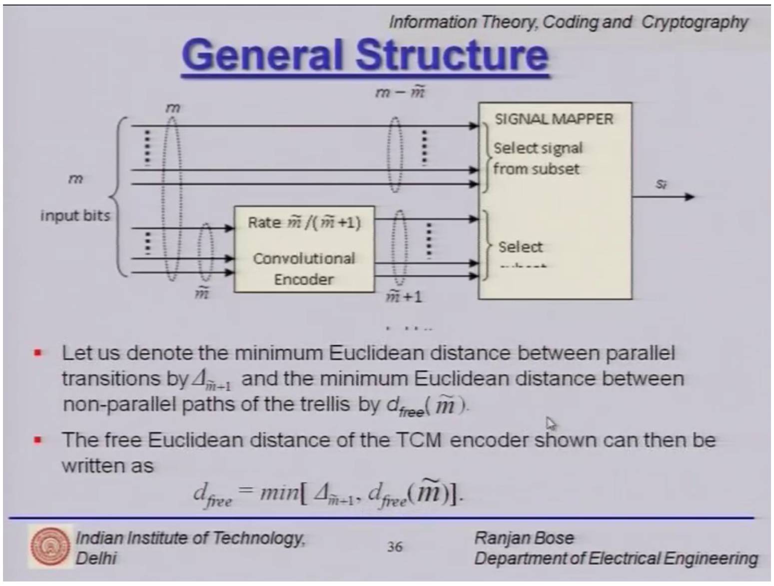
So, again consider the expanded 2 raise power m plus 1-ary signal set used for TCM we do not necessarily have to continue the process of partitioning till the last stage.

Suppose the desired Euclidean distance to be obtained just after the m plus 1th set partitioning step is m tilde less than m; then it can be seen that m tilde plus 1 steps we have got m tilde 2 raise power m tilde plus 1 subsets each subset containing 2 raise power m minus m tilde single points. So as you go down the number of signal points reduces and you have to only go down to those many points for which you have the branches to assign to so you do not have to go right into the bottom of the parameter.

(Refer Slide Time: 36:03)



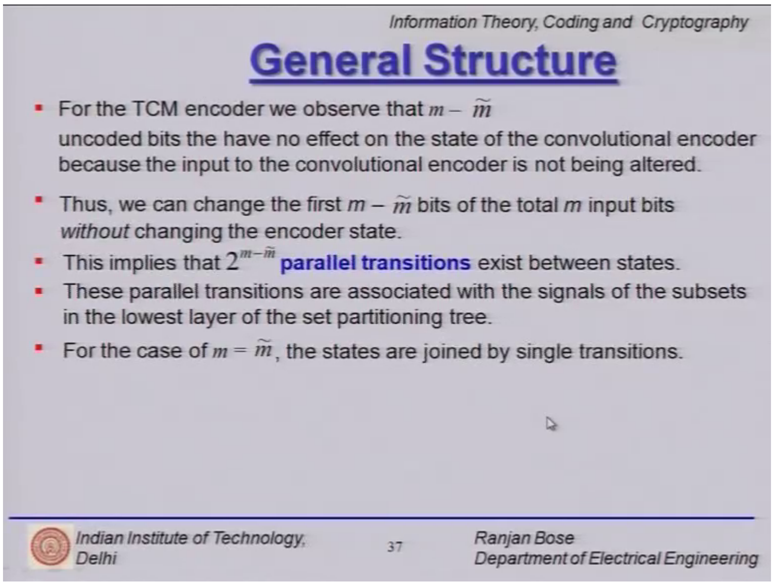
(Refer Slide Time: 36:07)



So, now let us look at a simple example of a general structure, suppose you have got m bits so first you have a convolutional encoder here it is systematic so you have m minus m tilde points going directly to the signal mapping and then m tilde over m tilde plus 1 convolutional encoder giving you the coded bit streams. The first m minus m tilde points is used to select the signal from the subset and these are used to select the subset so you have the subset and within the subsets you have signals.

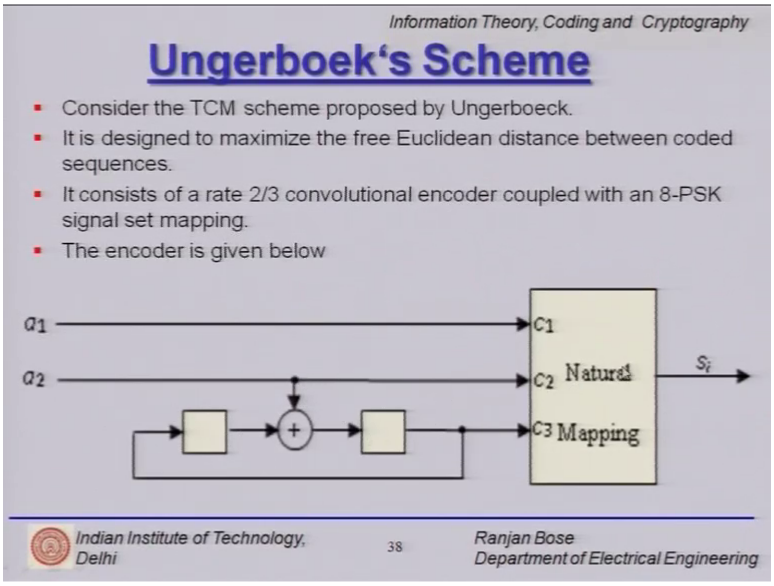
So, this is a strategy how to choose the subset and then a particular signal from the subset and this will give you a minimum d free in terms of the deltas.

(Refer Slide Time: 37:01)



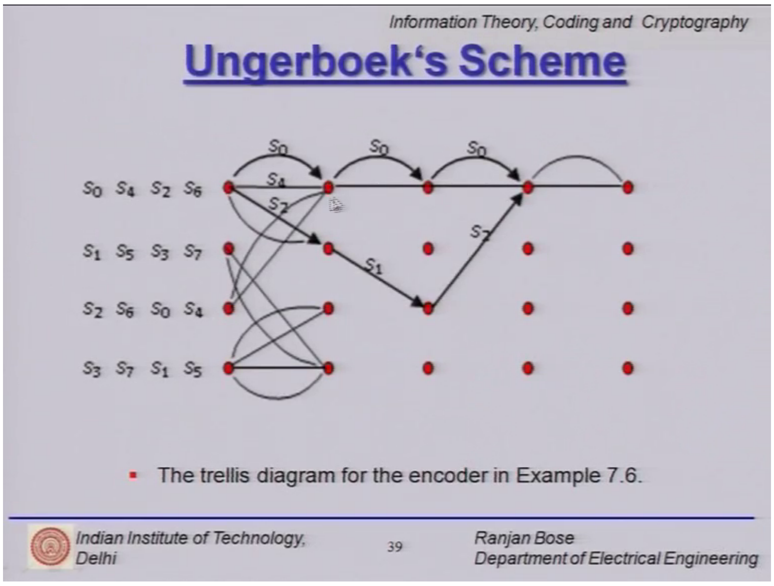
So, this is the general structure that we have just described how you can use a TCM encoder.

(Refer Slide Time: 37:08)



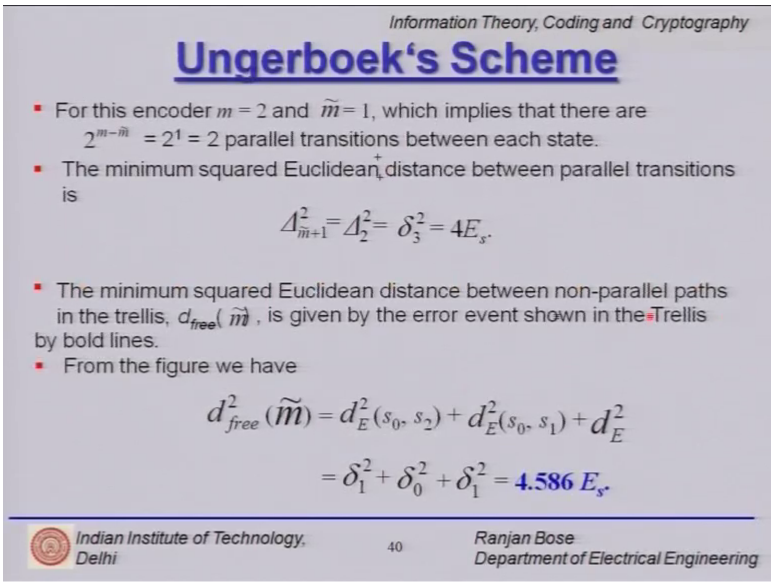
Now, we go on to Ungerboek scheme which tells us how to assign the symbols on to the trellis having done the set partitioning so the aim is to maximize the free Euclidean distance between coded sequences so just consider an example of a rate 2 by 3 convolutional encoder very simple 2 by 3 encoder natural mapping we have shown below.

(Refer Slide Time: 37:37)



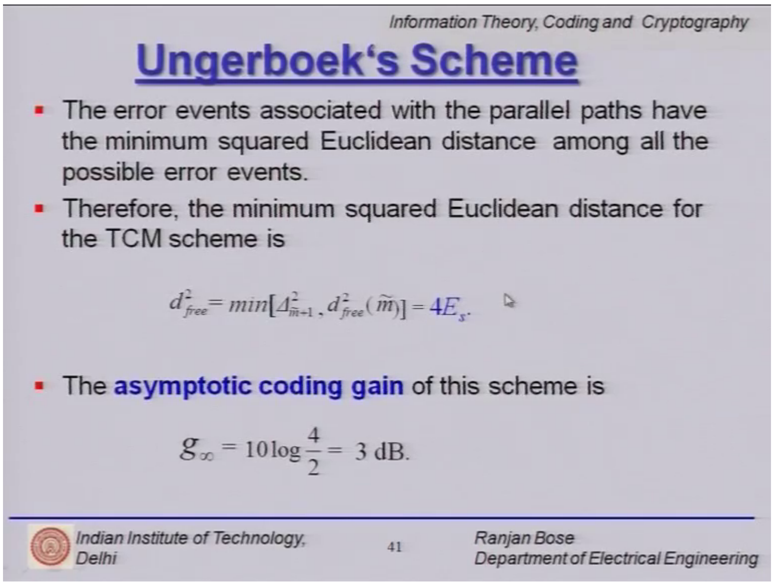
And for that we are going to do an Ungerboeck scheme and here you can see that there are parallel paths in this case because you know if this input a 1 does not change then the state transition does not change and for a 2 between 1 and 0 there are 2 possible outputs so there will be 2 paths from any 1 state to the next state, but the question is how to assign the symbols from the 8-PSK.

(Refer Slide Time: 38:13)



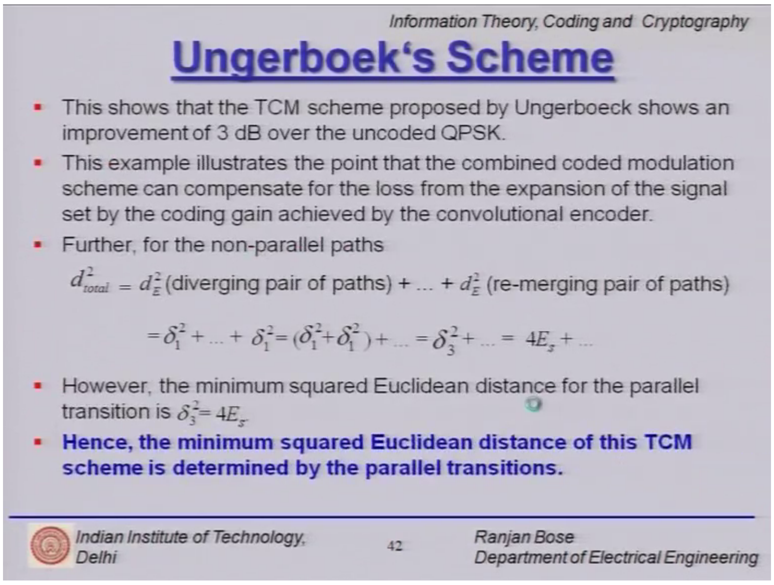
So, here we have used the Ungerboek scheme which we will discuss how to assign the different symbols from the;

(Refer Slide Time: 38:25)

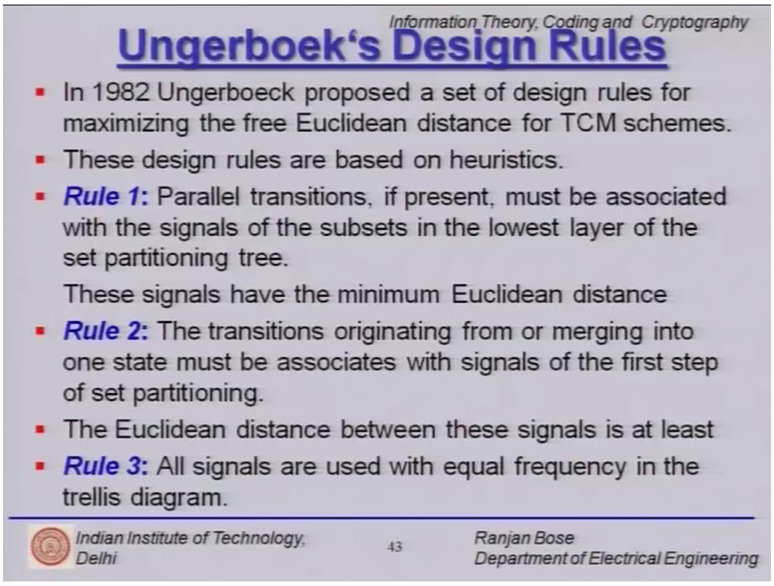


So once you do a mapping as described earlier then we can calculate directly the d free as the minimum of delta square m tilde plus 1 and d free squared m tilde which is E 4 comes from the maximally separated points on the symbol set and this tells you that the d free is larger than the uncoded one giving us an asymptotic gain of 3 db. So we have been able to recover and we have been able to give you a 3 db gain using a TCM scheme this called the Ungerboek TCM scheme.

(Refer Slide Time: 39:07)



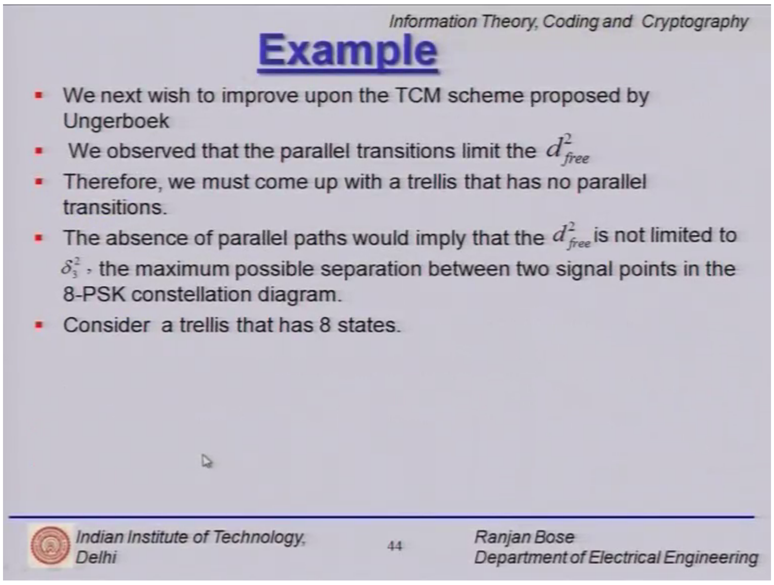
(Refer Slide Time: 39:13)



So, let us look at the design rules first and then revisit the example; so these were proposed in 1982 by Ungerboek these are heuristic design rules; what are the rules? Well they tell us how to assign symbols from the set partitioning to the trellis. So if a trellis has parallel transitions right then they must be associated with the signals of the subset in the lowest layer of the set partitioning tree that is they are maximally apart they have the maximum minimum Euclidian distance.

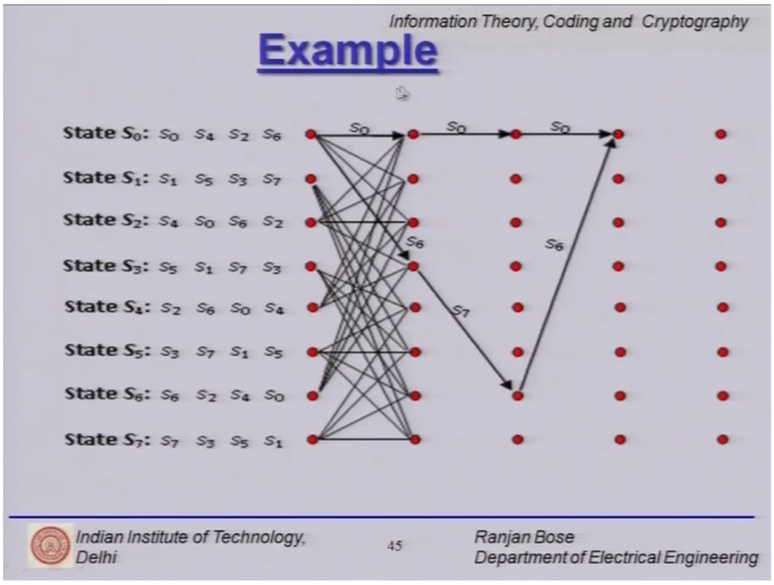
Rule 2 is that the transition originating from or merging into 1 state must be associated with signals from the first step of the set partitioning we already have done the set partitioning and then rule number 3 is all signals are used with equal frequency in the trellis diagram so these are heuristic rules which will ensure a good design of your trellis.

(Refer Slide Time: 40:11)



So, we want to improve the trellis scheme we just proposed and we want to improve the performance of that scheme using these design rules. So, let us look at a trellis with 8 states, let us look at an example where we will execute all the points proposed by Ungerboek.

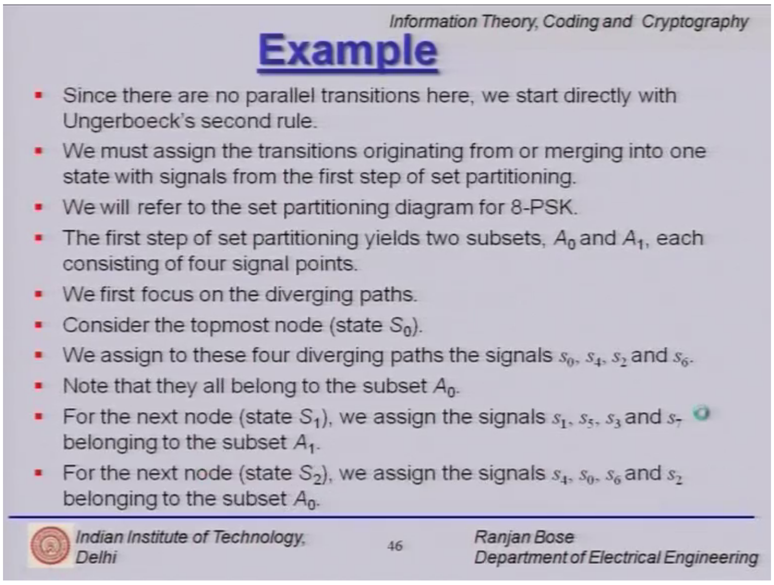
(Refer Slide Time: 40:35)



So, this is a trellis with 8 states and as you can see that there are 4 branches emanating from every node ok. So that 2 bits coming in for this convolutional encoder and there are clearly s 0 to s 7 so there are 8 points in the constellation diagram so 3 bits are being mapped to the symbols so it is a rate 2 by 3 encoded that has resulted in this trellis.

Since there are 3 memory elements therefore, there are 8 states in the trellis so the number of input bits is pretty much independent of the states in the trellis which depends on the memory unit and the number of output bits basically decide that the constellation size that we will use; so this is the trellis that we have at disposal.

(Refer Slide Time: 41:36)

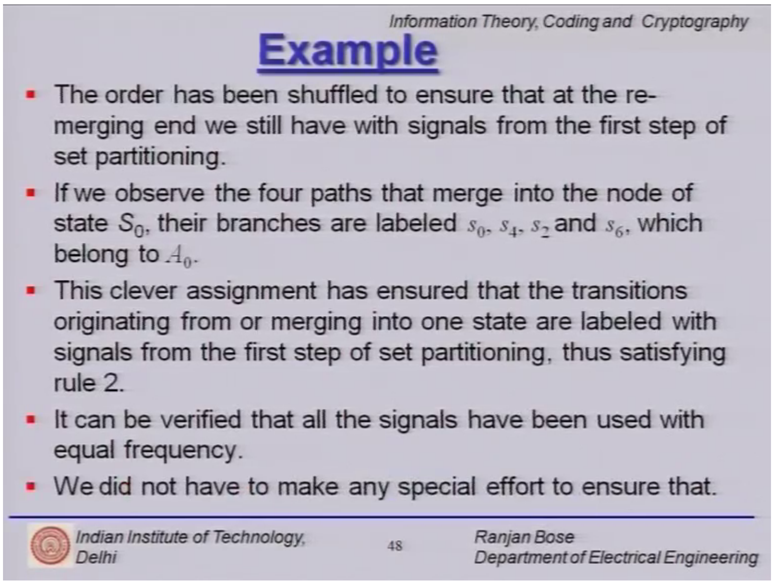


So, there are no parallel transitions we start directly with the Ungerboek second rule of assigning; so in this trellis we have assigned s 0 s 4 s 2 and s 2 from for these 2 for these 4 branches which are diverging out.

So, this is from the lower most but 1 level of the set partitioning that we have done already. Similarly we have taken the next one s 1 s 5 s 3 and s 7 these are the other 4 maximally separated points in the trellis and so and so forth and we have make sure that these are assigned with equal frequencies. Same care has been taken for the diverging and the merging back paths; so if you look at the points which are merging back they have been also assigned with that same care.

So, in this slide we are looking at the diverging path and this slide describes why s 0 s 4 s 2 and s 6 symbol have been assigned to that they belong to the subset a 0 and for the next node we have done s 1 s 5 s 3 and s 7 as we just not discussed ok.

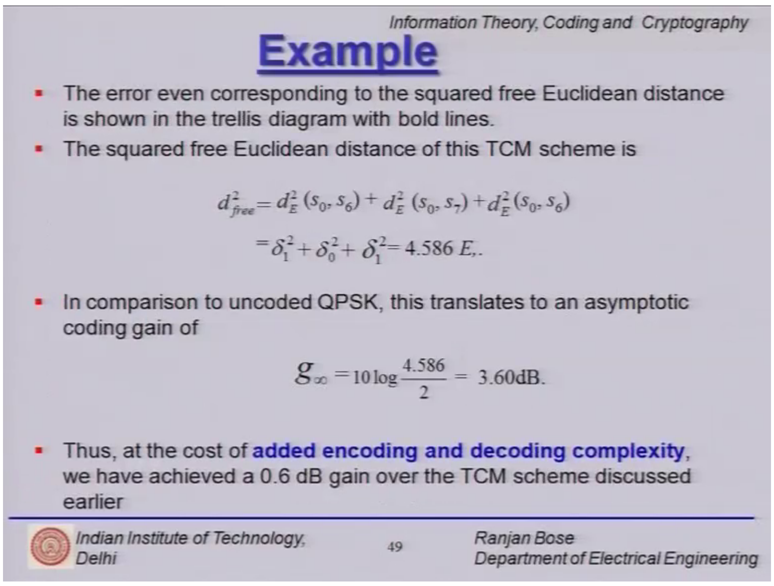
(Refer Slide Time: 43:05)



So we have done all of this and the order has been shuffled to ensure that the re the at the reemerging and we still have the signals with the first step of the set partitioning that is the care we have taken to do so and we have also ensured that the all the symbols have been used with equal frequency.

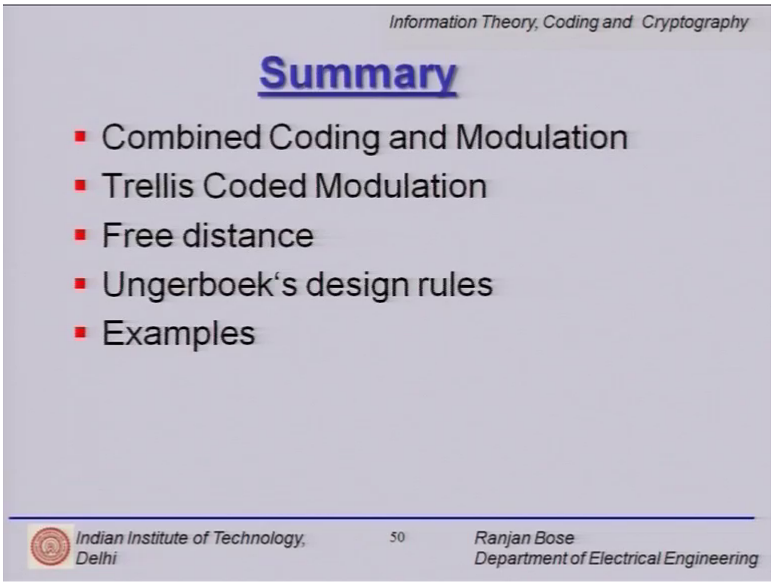
So, we have been able to use Ungerboek design rule to assign symbols to this trellis diagram with 8 states.

(Refer Slide Time: 43:31)



Lets quickly calculate the d free, so d free we had already highlighted the 2 paths which result in d free and the d free calculation gives the g infinity 2 3 point 6 db therefore, this has a even better point 6 db gain over the TCM scheme discussed earlier by Ungerboek.

(Refer Slide Time: 43:56)



So, let us now kind of conclude this lecture what we have looked at is the notion of combined coding and modulation then we introduced the idea about trellis coded modulation and how we can get improved performance without additional power or bandwidth then we talked about the free distance and the Ungerboek's designed rules we also followed it up with some examples with that we come to the end of this lecture.