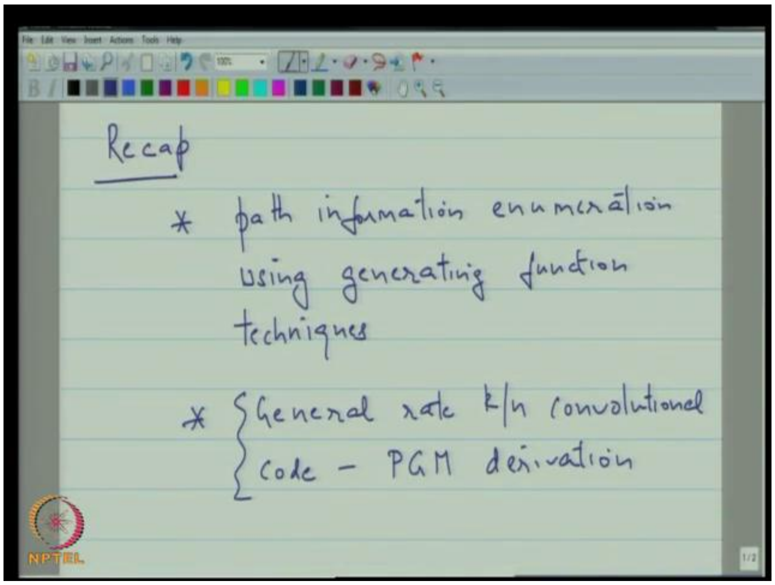
Error Correcting Codes

Prof. Dr. P Vijay Kumar
Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore

Lecture No. # 20
Viterbi Decoder Over the AWGN Channel

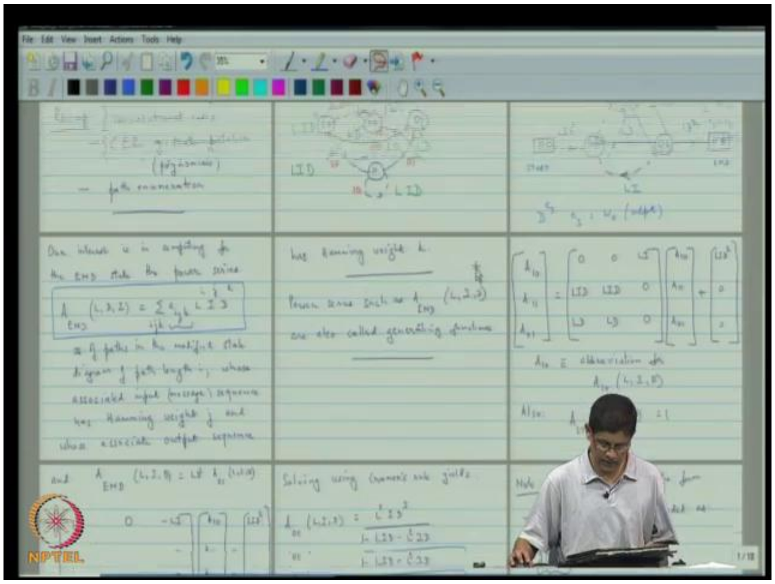
(Refer Slide Time: 00:39)



Good after noon, we will start our twenty lectures. Today and so this will be the last segment on the convolution code for now, before we move on to a new topic. Just to recap what we did last time, we spent most of the time in the last lecture on what I call path numeration; and the idea here is that you wanted to look at the trellis of the convolution code, and you wanted to actually, extract information about the various code words in the code, each of which corresponds to the path in the trellis. So that is why we call it path enumeration.

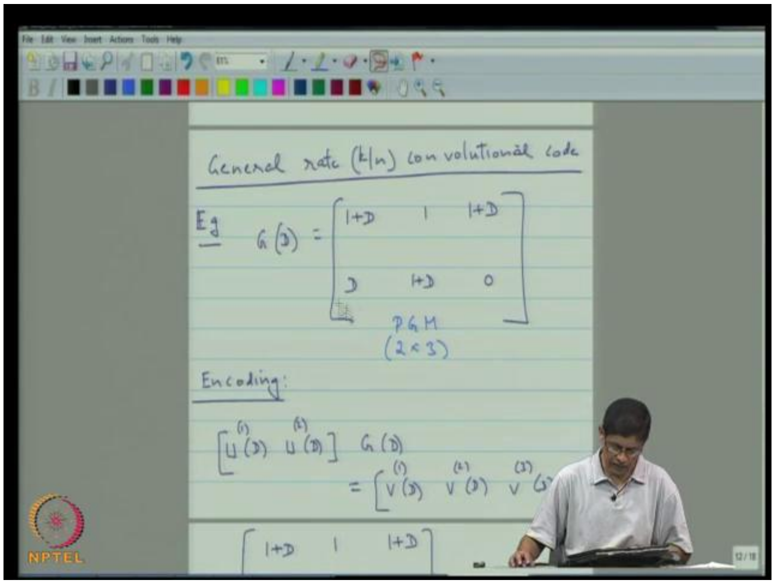
And for each path we associate a length of the path, which you can think of us right now is the number of messages symbols for the rate one by n code or is the number of edges, that is travels by the path, and then we get track of input hamming weight, that is a number of message symbols, that are equal to one as well as output hamming weight, which is hamming weight of the code word itself. And also we derive the polynomial generated metrics for a general rate, k by n convolutional code.

(Refer Slide Time: 01:34)



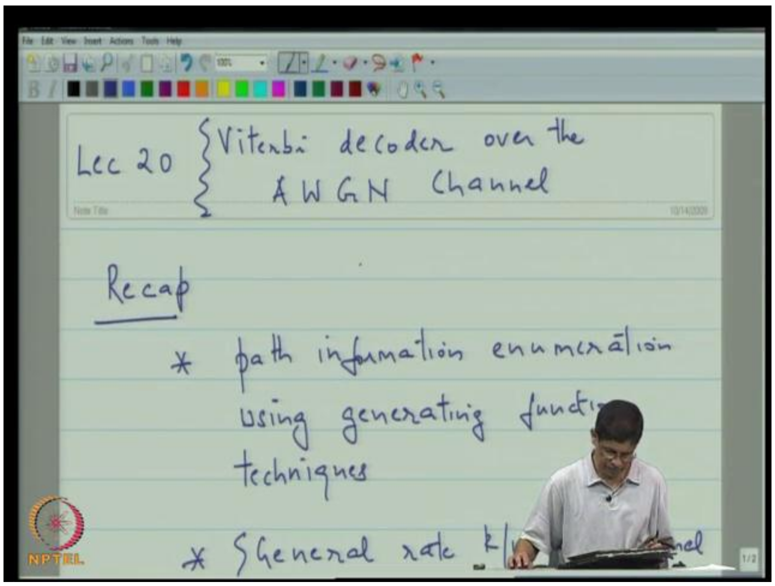
So perhaps, I can just quickly run you through the last lecture here. So here this part we use the modified the final state machine, and the modified trellis to actually derived the generating function, which actually gives you all the information about the various path and the information is in the form of is in the form of power series where the coefficients tell you how many path links certain information weight certain hamming output weight.

(Refer Slide Time: 02:20)



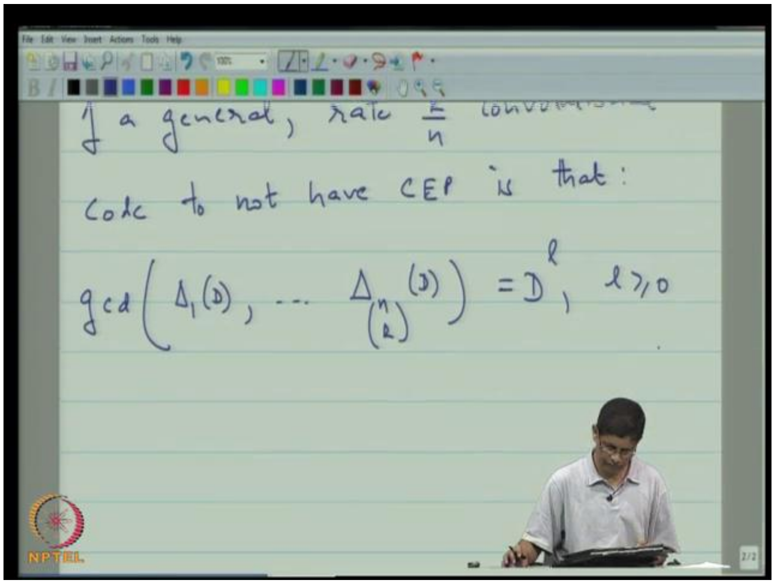
And then after that, we went on talk about general rate k byte and convolution code, where went to was in strange and within this one example, and then we took up the general case with for this example here what we did was we took polynomial generation metrics and then we do the encoder. Then we said what will general rate k byte convolution code look like, and we wrote down on a equation that will discuss generation of the code symbols. And after that I went on to actually, deriving the polynomial generated metrics right.

(Refer Slide Time: 02:58)



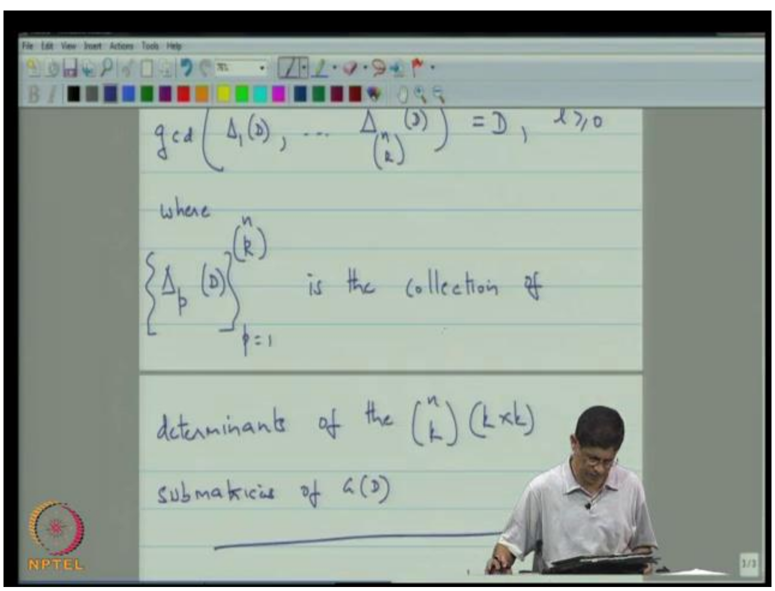
So that is where we left of last time. Today, what we will do is, we will talk about the Viterbi decoder over the additive weight Gaussian noise channels that is our principle goal. But first let me just get one other matter out of the way; if you will recall then we were discussing rate 1 by n convolution code, then there was the issue of catastrophic error propagation. And there we derive the condition to avoid catastrophic error propagation or at least avoid having generated matrix or p g m that catastrophic error propagation.

(Refer Slide Time: 03:40)



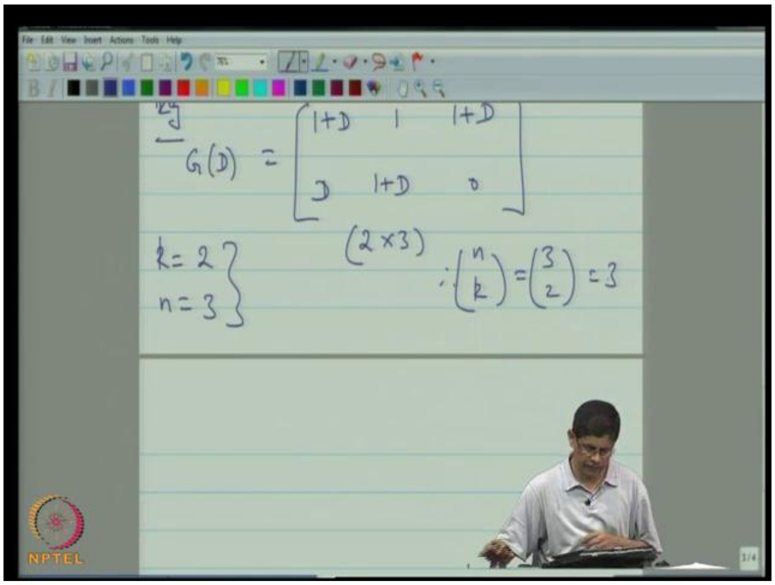
So, the analysis result propagation rate k by n code and going to actually, put that down. Theorem analysis putting bracket CEP, a necessary and sufficient condition on the P G M of a general rate k by n convolutional code to not have case tropic error propagation is that the GCD of delta one of D comma delta n to is k of D is equal to D to the l for l greater than or equal to 0.

(Refer Slide Time: 05:02)



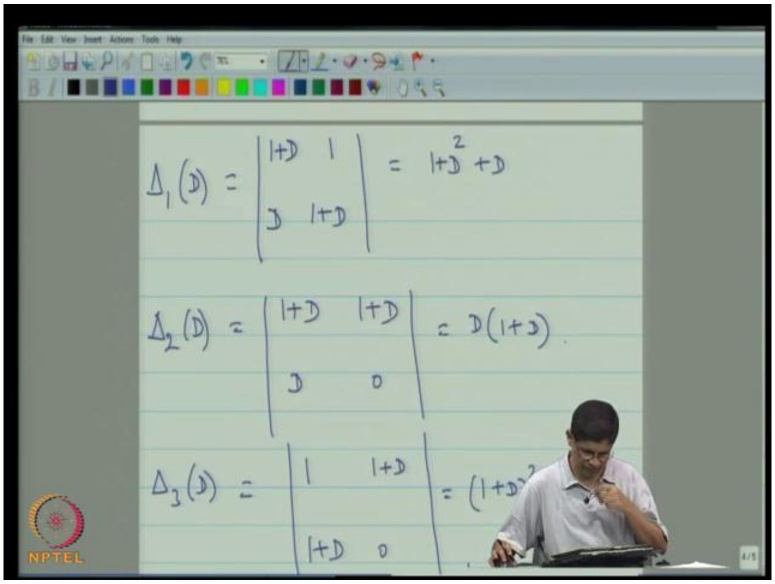
Where delta p of D, the collection has p is varies from 1 to n is k, is the collection of determinants of the n choose k k by k sub matrices of G of D. Let zoom of necessary and sufficient condition is that we look at all the determine of the sub matrix is, then that GCD will be D to the L. I think become theorem to look the example.

(Refer Slide Time: 06:30)



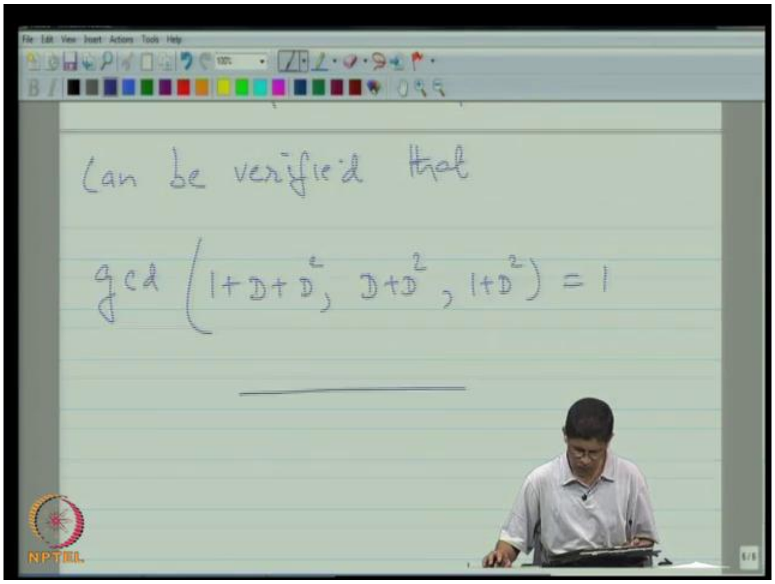
See supposing G of D, is the same code that we discussed earlier which is 1 plus D one plus D D 1 plus D zero, and now this is as you can see 2 by 3 matrix.

(Refer Slide Time: 07:05)



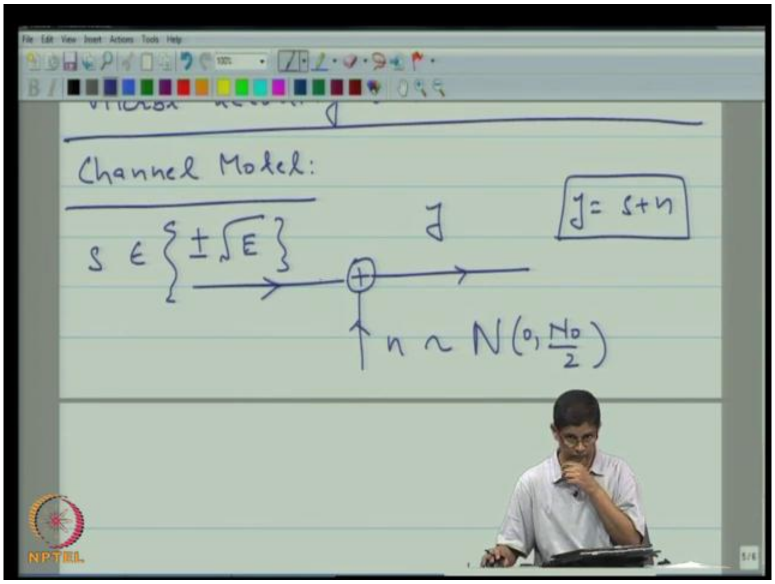
So, in this case k is equal to 2, n is equal to 3. the determinants of all the k by k summit to result D of D. That means in this case n choose k here, therefore n choose k is equal to 3 choose 2 which is 3. So, there are 3 of matrix is, so delta one of D is determinant 1 plus D 1 D 1 plus D which is turns out to be 1 plus D square plus D del 2 of D is determinant, so here I took first column in the second column. Now, take the first column in the last column. That it will be determinant 1 plus D 1 plus D D and 0, which is D into 1 plus D in delta 3 of D is delta 3 of D. Now going in to take second in the third column, so it is a there will be 1 1 plus D 1 plus D 0.

(Refer Slide Time: 08:58)



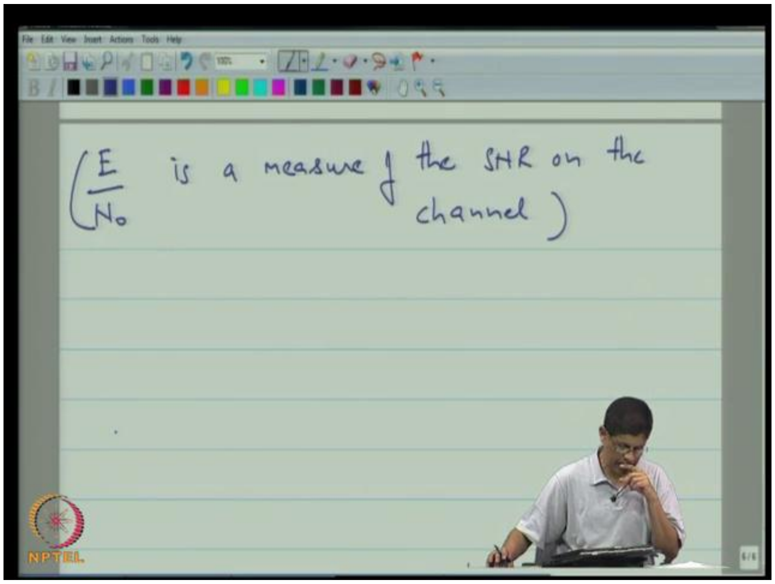
So this is 1 plus D the whole square, now you can verify that the GCD of that the GCD of 1 plus D plus D square D plus D square 1 plus D square is actually equal to 1, and now to asked how do you find GCD when there are three, we saw how do they when there are 2 what you have you can find theory of two. And then find GCD of third with GCD of first 2 and how you break there are does not matter, but see in the particular in sense, because if you add first and the second you get one, so anything divides in the second must divide 1. The GCD has to be one, so it is value clear. So, that is value you apply there.

(Refer Slide Time: 10:31)



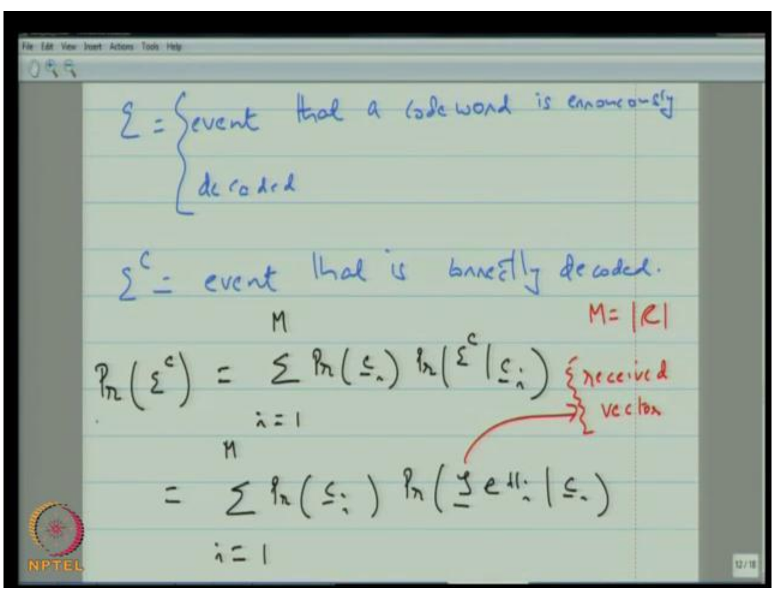
Now, other topic in convolution codes, this is the title of lecture is how do you do Viterbi decoding over the auditing white guys in noise channel, so here for channel model will be like this, and we will be the work with this is in the channel model in this case. The input has belongs to plus or minus square root of E, the noise n is distributed as 0 N 0 by 2 and the output here has y and y is equal to s plus n. So this will be are module, and of course this is for single channel use what will be doing this will be using the channel over and over again, and for one used to the next excuse me wewill assume that in the noise samples are independent.

(Refer Slide Time: 12:40)



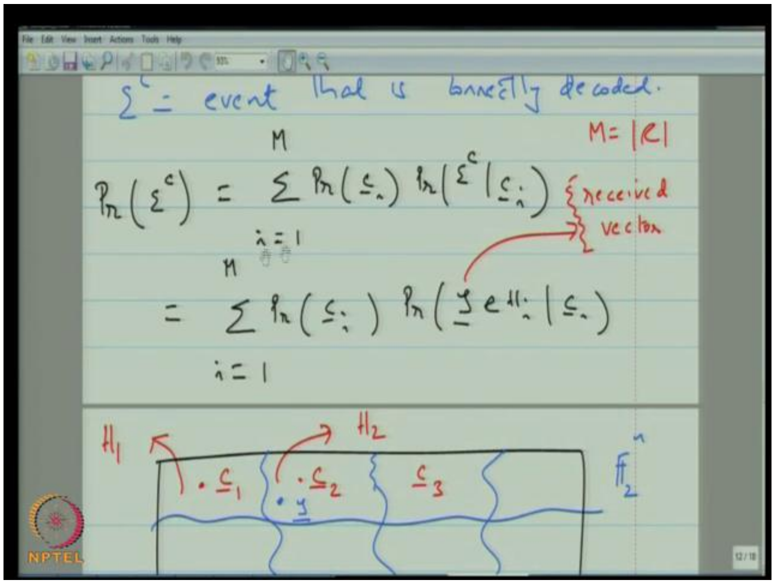
We will assume E by N 0 is a measure of the SNR on the channel, and some time back will be discussed maximum likelihood decoding over band width symmetric channel. We said the minimum probability of error decoders some bandwidth symmetric channel, we showed that envelope has been minimum hamming distance decoding. There is here, so let me see if I can quickly pull that up from a earlier lectures.

(Refer Slide Time: 14:17)



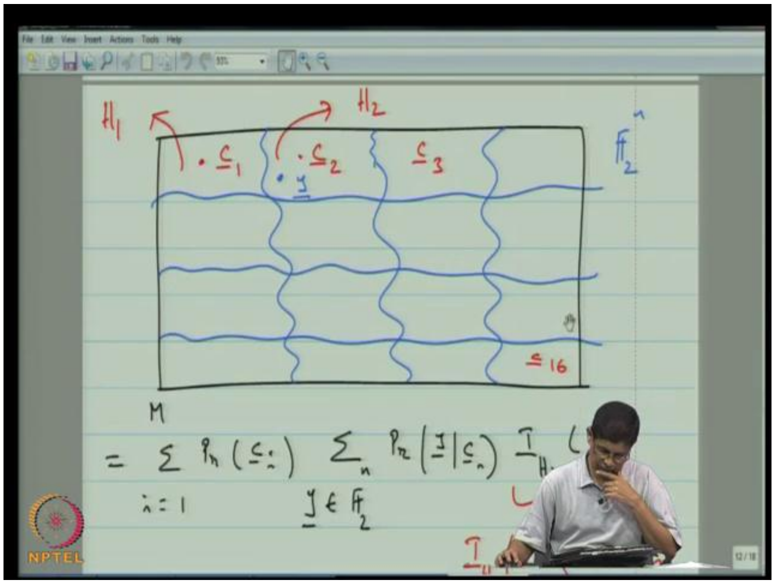
Here will discussing the minimum probability of error decoder which was the band width symmetric channel, and we wrote on expression for probability that correct the code word is correctly decoded.

(Refer Slide Time: 14:30)



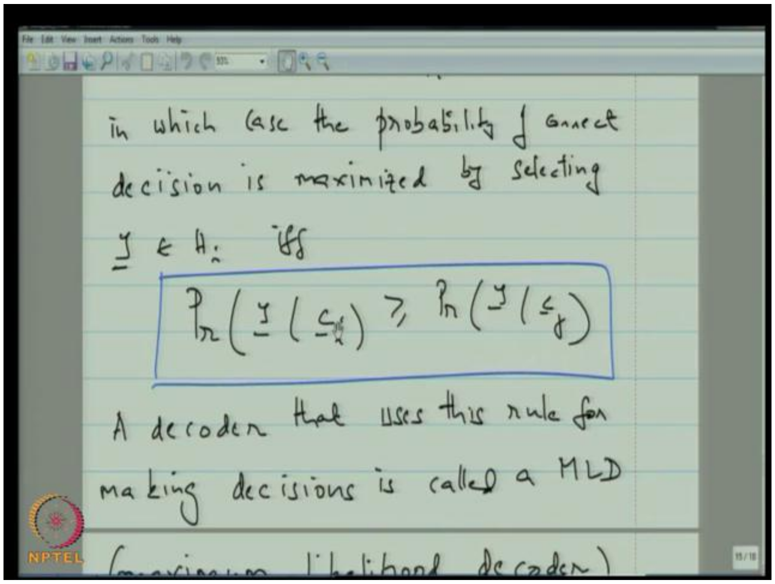
We said about trying to design the decoded in such way this probability is actually minimized that you minimized the quantity here, and we founded that decoded that minimized this quantity maximum likelihood decoder, that is for any given vector y choose that code word such that this corresponding like conditional probability function is logiest, so this called maximum likelihood decoded.

(Refer Slide Time: 15:05)



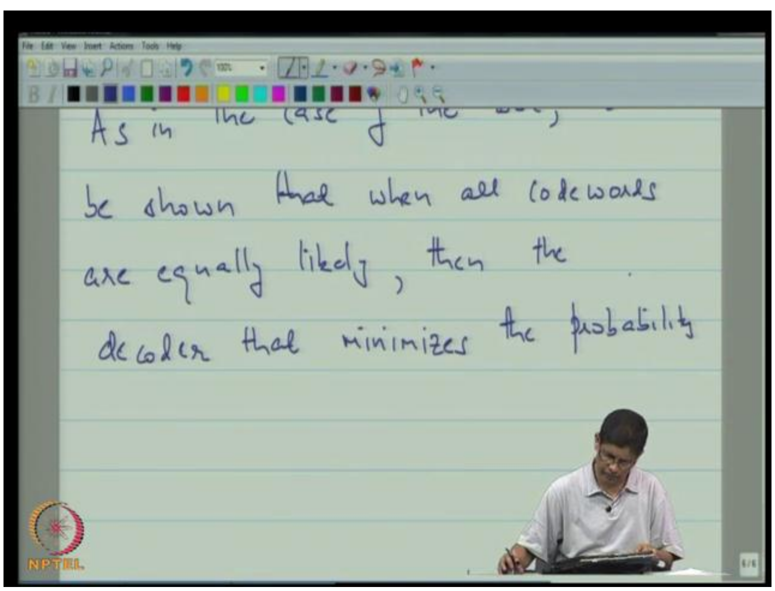
Now, in the case of binary symmetric channel here, we went on to with derivation and in the derivation here, we actually recognizing that the vector that we see over the binary symmetric channel belongs to a discrete and sample. We are a summation if y is drawn from continuous on sample then the summation will be replace by integration sign, apart from that the derivation is exactly the same, even channel. And again the conclusion you also the same that if you assume that a code words are equal likely, then the minimum probability of error decoded is just one there actually choose the code word that maximize this.

(Refer Slide Time: 15:50)



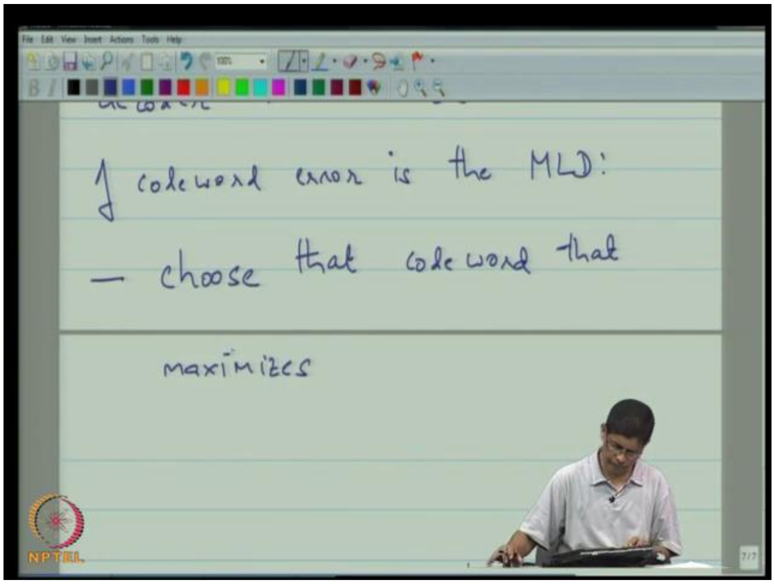
Now, this was the probability mass conditional probability mass function for binary symmetric channel. probability joint probability conditional density function, conditional density function other than that there is no difference.

(Refer Slide Time: 16:24)



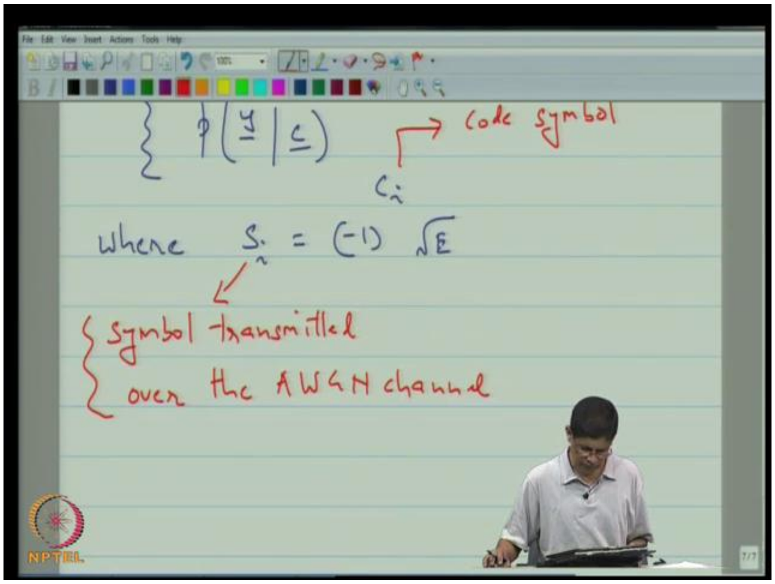
Let us go back here as in the case of the bandwidth symmetric channel, it can be shown that when all code words are equally likely, then the decoder that minimizes the probability of code word error is the maximum likely decoded.

(Refer Slide Time: 17:47)



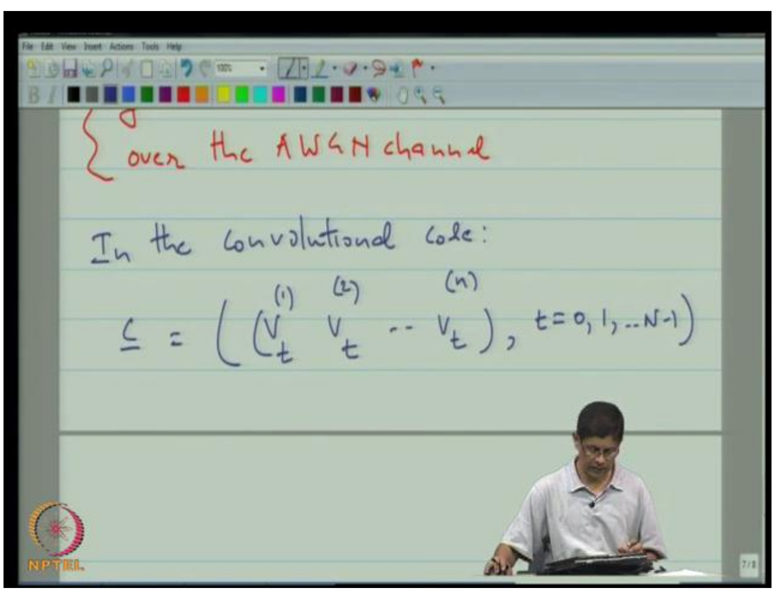
So, where actually says the algorithm, that choose that code word that maximizes probability of y given the code word. Now, just one common word here.

(Refer Slide Time: 18:53)



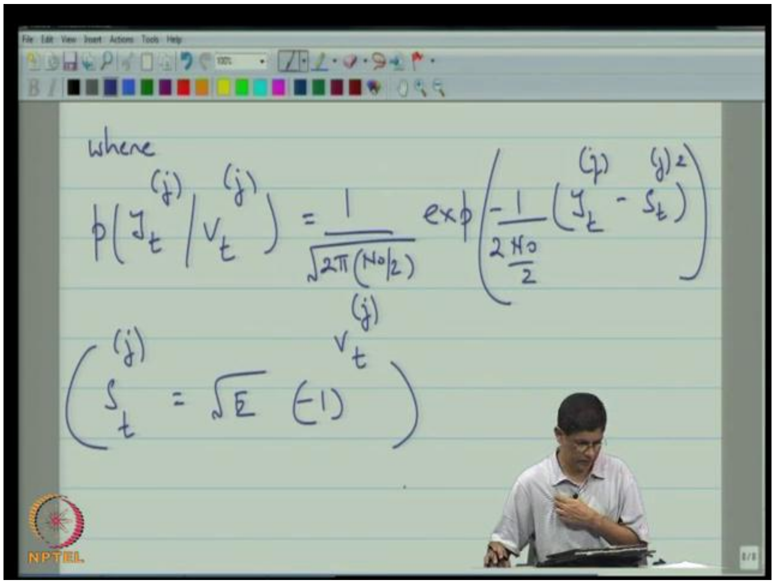
When round the channel module here, I put down input s which is plus or minus, but or coders always been 0 or 1. How do you make the connection; the connection is enough, where S i is equal to minus 1 to this c i times root E this is the nothing, so this will be here is the code symbol whereas this here is the symbol transmitted. You know the clear that talk let us see if can with more information about the decoded.

(Refer Slide Time: 20:33)



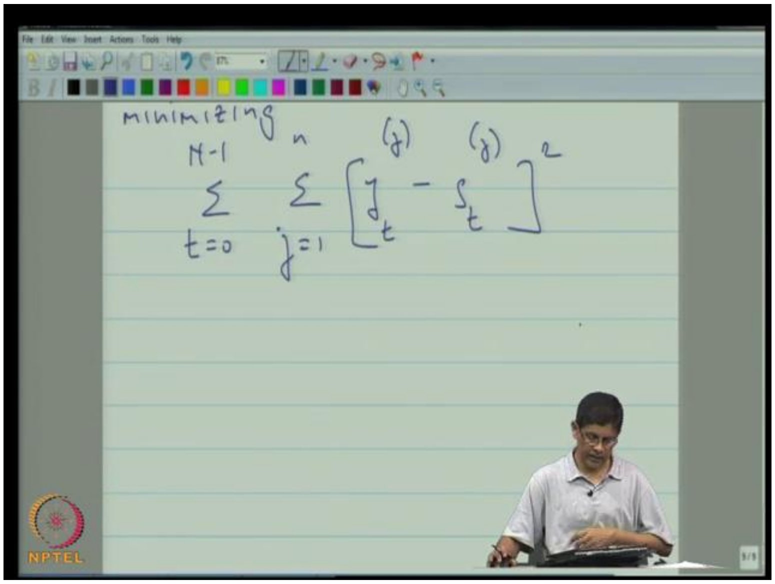
Now, in the case of convolution code, here we have seen here code c is really the collection of vectors.

(Refer Slide Time: 21:27)



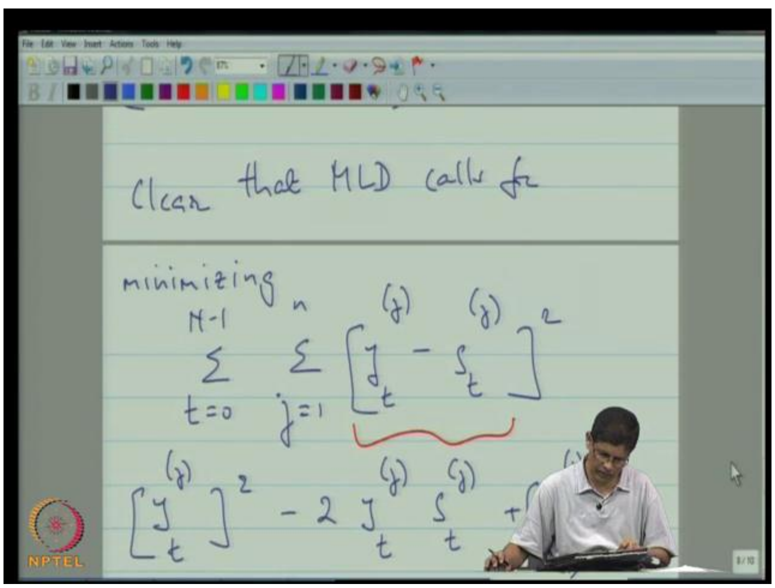
So this is what here code word will look like, it will be the collection of all interrupt output. So then in this case p of y given c breaks down to the product t is equal to 0 n minus 1, the product j equal to 1 to n p of y t j given v t j, and where p of y t j given v t j is 1 upon root 2 pi n 0 by 2 exponential minus 1 upon 2 n 0 by 2 y t j minus S t j whole square. And S t j is minus 1 to the v. And then given some as space to write that t j is root E minus 1 to the v some j of t, so then it is assuming so that you get a prospective. So, maximum likelihood code word decoding, as we to actually select that code word which maximize that p of y given c, but p of y given c is really, the product of several terms each of which actually looks like this. Now, it is clear that if we want to maximize this quantity what you really meet to do is minimize this exponent, because of the negative sign you want to minimize this; these are constants which do not affect the maximization, we want to actually minimize this distance.

(Refer Slide Time: 24:18)



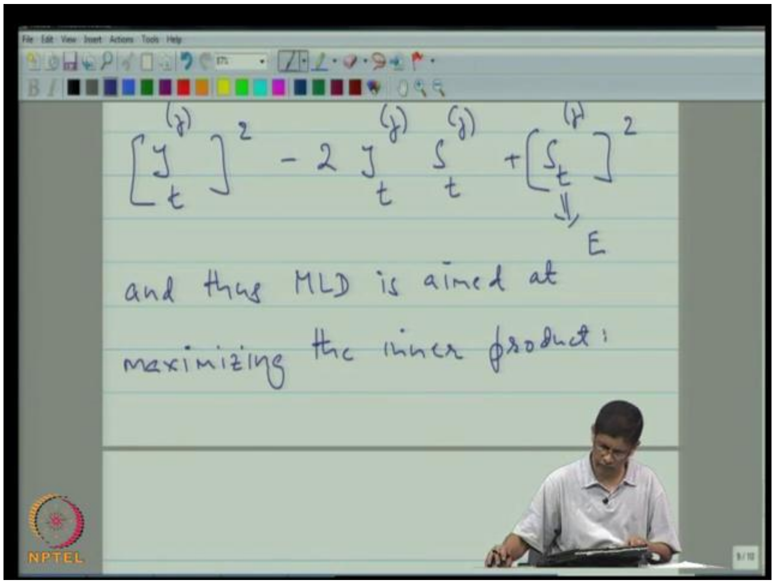
So, it is clear therefore that maximum likelihood decoding calls for minimizing the quantity the sum t is equal to 0 to n minus 1 j equal to 1 to n y t of j minus S t of j the whole square.

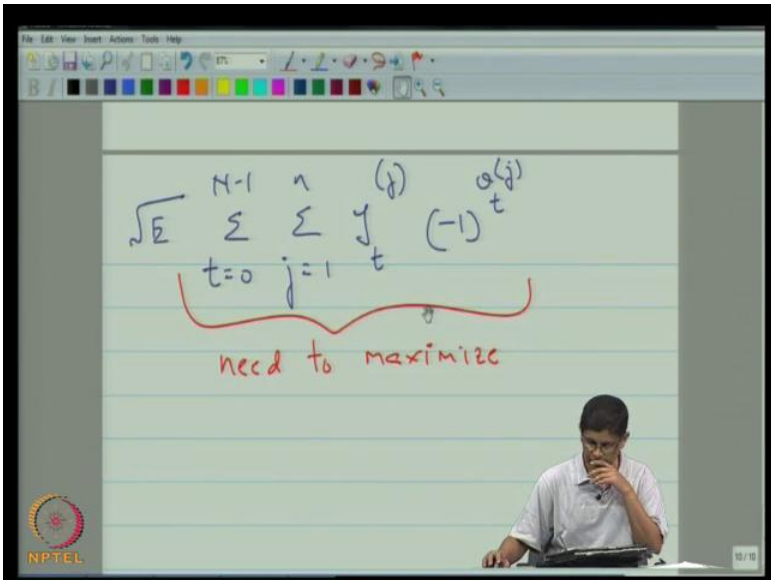
(Refer Slide Time: 25:01)



Now, this in term in the expand it in to y t j square minus twice y t j s t j plus s t j square, we want to minimize this is your interested minimize this quantity. But this term are constant upon the code word because this maximization that trying to differentiate 2 code words. But now here, we term which is depending upon code word free to ignorant we can drop this term. Now, on other hand in similar on a similar note this think really, if we look the expression for t if you for as t always get E here, some this expression here if you square this quantity the depending on the codes disappear this get E. So, again you get a quantity that is depending on the code word and therefore can be dropped this is s t e.

(Refer Slide Time: 26:30)

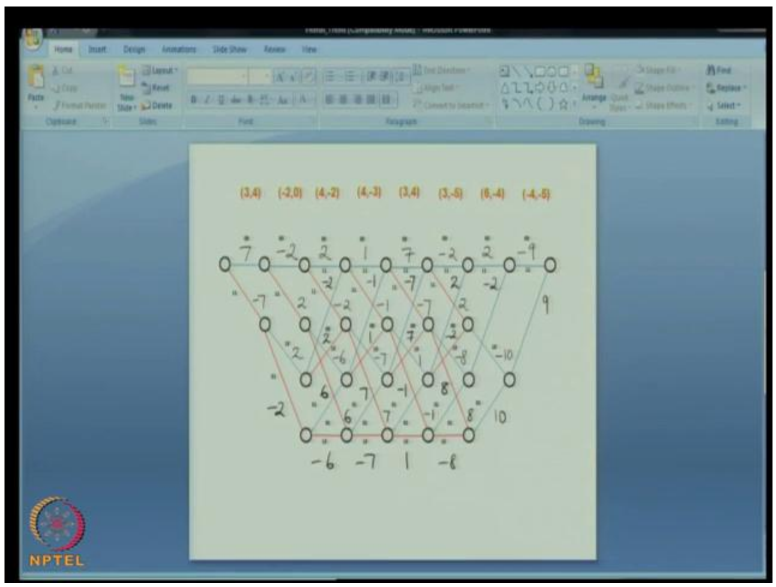




So this can be dropped it is clear that from there is and thus maximum likelihood decoding is aimed at maximizing again, the inner product which is square root of e. The term t is equal to 0 to n minus 1 j equal to 1 to n y t j minus 1 to the v minus 1 to the v t j. Again there once again this consequent root is no consequent, so then this is what we choose to minimized or choose to maximized.

What we want to you recognize this is really in a product between 2 vectors; one is the stream of symbols wise up t j and other is a corresponding codes in this plus and minus in the format. We want to take a inner product and actually choose that code word, which actually maximize this in a product.

(Refer Slide Time: 28:36)



Now, let see we can actually look at the competition on the trailers, and I am going to try something new here, and I am going to try to duration in pop points. So, here this now is a same trailers that we had earlier when we have dealing with the band width symmetric channel, and we have actually carried out decoding over the band width symmetric channel, we can drawn a trellis we can actually look at the decoding. So, this is the same trailers the same code. The only differentiate here is that, now what we have what have put down in the top for the 2 receive symbol corresponding to each branch of trellis; these are the received symbols y t 1 y t 2 for t equal to 0, this is y t 1 y t 2 for t equal to 1 and so on. And in this case, we have 1, 2, 3, 4, 5, 6, 7, 8; so in fact n is equal to 8.

So, we are looking the decoding code words whose branch length on the trellis 8 that corresponds to 8 necessary symbols, and 6teen output symbols. However, if you must keep 9 in their this is the terminated trellis, because here we start in the trailer in the state become in fact all 0 state then we back to the 0 state in last to the maximum symbols are 0 and thus do not contain information. There are 6 basic symbols but I there want to show you here is simply function of decoder for this case now keep in mind that all that what we do as was take the inner product between plus minus convolution of code symbol and received factor. Now plus minus 1 now written in very small plunderer 0 is to allow me keep track calculation.

You may not see in your screen there is same trailer as before the 0 here, translate in the plus minus calculation is one when you take in a product one one in 3 4, this is 7 now to right bound the corresponding every one of the what we have aiming for is the inner product between plus minus samisen of the code word entire received factor but actually, it is clear at branch at time clear from the nature product. But you can actually, complete inner product for branch all the branches making up path that will get inner product for the entire path right; that is clear that is what we are going to do, so I am just going to simply, write bound this inner product easily take any colour.

I guess, I should take red and I am going to pick a ball point pen. Let us see if I back put down now here, the 0 corresponds to one that is 3 plus 4 that is 7. I guess a colour invokes that is 7 and for the next symbol. Similarly, minus 2 plus 0 that minus two, I see in screen not really show in a very well. So path that which good idea trial let me get back to let be erase this and get back to back will state get back and the ball point pen and put down the numbers and note will come out this is then 7 this is minus 2 2 1 7 minus 22 minus 9 and I want to write corresponding numbers for all the branches in trailers next look at the one one branches.

One one branches or corresponds to plus corresponds to minus 1 and plus minus 1 in the format so it is minus 3 minus 4. So, I have to write down exactly same number as earlier except there are flip design these are this will be minus 7 and looking at like good place put that down would be here, so that is minus 7 that is 2 that is minus two minus 1 minus 7 2. Now, I want to put down the corresponding numbers corresponding to this 1 0 7 1 0 means minus in plus. What means there it take the second symbol as it is but the first symbol I change the sign. Then that is leads to become 2 this will become minus 6 minus 7 one minus 8 and minus 10.

So there are 8 branches I have 5 may to go now looks that this branch is 0 one branch with such take the first one is the plus and second one with minus. So here this will be 3 minus 4, so I put down minus 1 minus 2 any trailer again here, I need I have wrong branch here, I need to take the first this is 0 1 branch. I need to take the first symbol of plus second with minus. This is minus 2 and this one is 6 this one is 7 this is that one is minus 1 minus 1 this one is 8, and often now let is look at branch in the bottom of trellis 1 0 that tells you that you take first negative sign there is minus 6 minus 7 1 minus 8. And then we have this 1, 2, 3, 4, 5, 6, second and here which correspond to have one these would be negate with sign that this would be minus 4 plus 2 which is minus 2 minus 4 plus 3 which is minus 1 minus 3 minus 4 is minus 7 3 plus 2 is 5 2 plus minus 4 is minus 2 we will now draw for I think most of the outputs ah there is there are 2 parts, which are 0 one here, so this 7 and this 6.

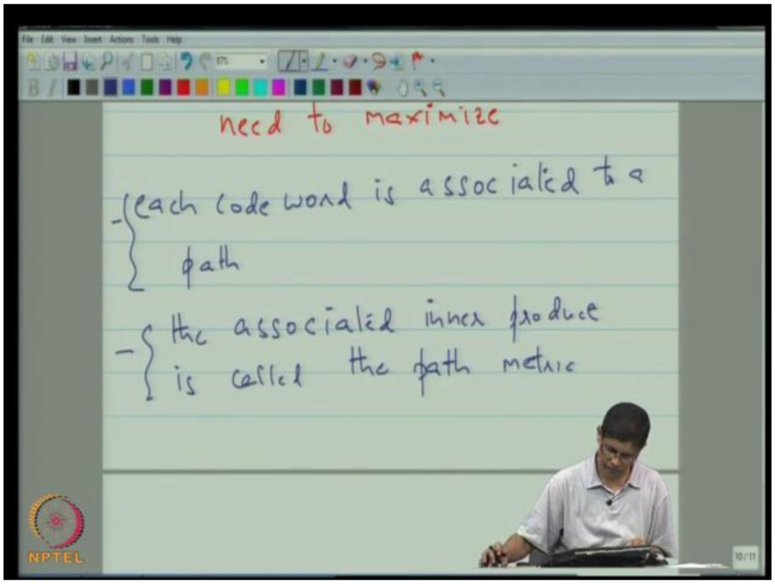
I will repeat as kept write down the numbers here quickly, this is also 6 also 7 minus 1 8 and here this is as 0 1, that would be 6 plus 4 this should be 10 and this one would be 9 I think only there are not include this 0 zero segment like in fill that easier that is 21 7 minus 2, it is look like ah living complicated, but all there are want to say that look at actually this trellis. Then all the inner product of associate the every code word is can be computed from the numbers there actually show here, but what you can actually you can proceed has in the case of Viterbi decoder you can proceed as in the case of the Viterbi decoder. Let see again back in the arrow back you can proceed as in the Viterbi coder for the band width symmetric channel. What you do as you go for left to right you can try to look at the code word that maximizes this in a product.

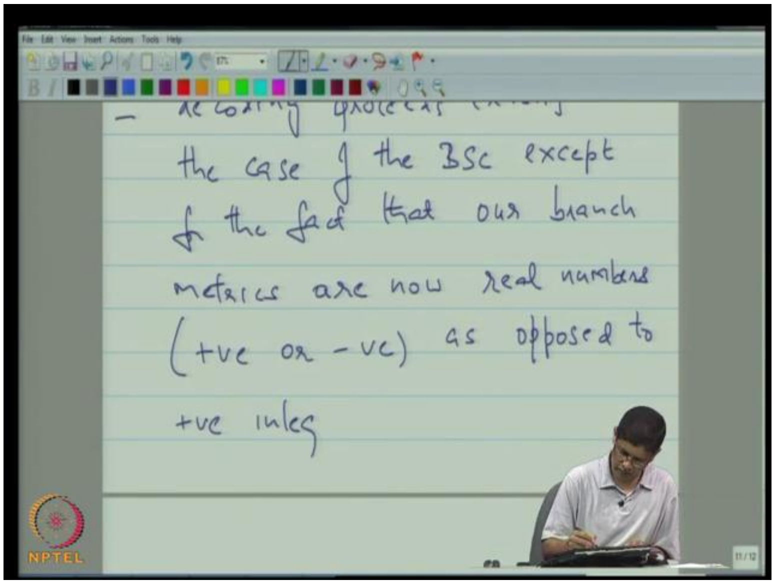
We start in here this is 0 and now, we have 7 and minus 7 these matrix are also transfer to this nodes then thereafter to accumulate metrics this is 7 minus 2 that is 5. So here the matrix here is 7 plus minus 2 5 this is 7 plus 2 9 minus 7 plus 2 minus 5 minus 7 plus minus 2 minus 9. Let you can actually do is a, you can keep track of metrics this is called metrics.

This inner product partial product is called branch metric, and total inner product code word metric you can complete or path metric, you can complete path metric or in the branch metrics and just like in the case of bandwidth symmetric channel. The real progress is made in the code in when you arrive at node where there are 2 containing paths, and you can choose between one of them. For example, here the total metric 7 minus 2 plus 2 which is 7 or the other choice which is minus 7 plus 2 minus 2 which is minus 7 your choice between 7 or minus 7 or of course will take 7, that is the this point to have a looser in the sense, that there is no way in code word which begin the trailers like this can never gone become code word that maximize the metrics you can actually discard that.

Now you see that once you economic this number and you so earlier the numbers there are integers there are positive, and you looking to minimize them metric path matrix here the numbers are real numbers and they could be plus and minus. And we have looking to maximize metrics. But happened that decoding proceed exactly same is before you can actually, go yet choose given in path just by going the section by section, and every node what you do as discuss at every node level will you discard one path reach node; that is for the complexity say in comes in but I think that given displaced not very clear eligible on 7 perceive any further, how will be the meaning as clear and will discuss on the decoding on this, it is very similar to what we did for the band width symmetric channel.

(Refer Slide Time: 42:35)

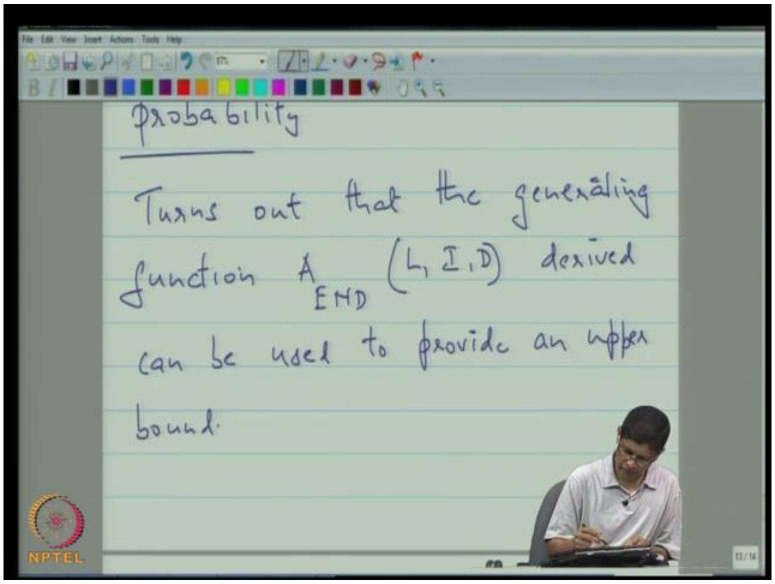




I just common to company what I just told you each code word is associated to a path the associated inner product is called the path metric, each path metric is the sum of branch metrics. And decoding proceeds exactly as in the case of the band width symmetric channel. Except for the fact that our branch metrics are now real numbers either positive or negative as oppose to the integer as opposed to positive integers in the case of binary symmetric channel. And finally, we are looking to maximize path metrics and not seeking the path with minimum with minimum metric as in the case of the band width symmetric channel decoding precise exactly in the case of band width symmetric channel except for fact there are branch metrics.

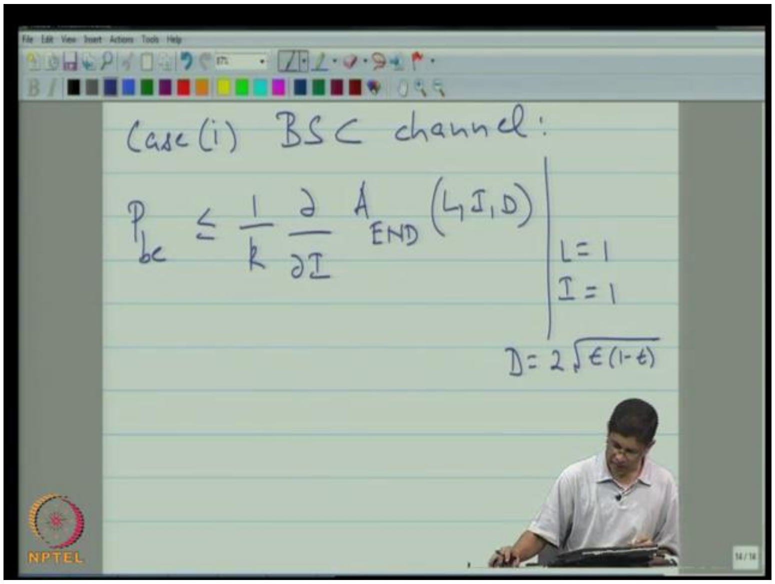
Now real numbers in a positive or negative as opposed to positive integer in the case with band width symmetric channel also, we are looking to maximize path metrics and not seeking the path with minimum metric as in the case of the band width symmetric channel .So will you go back and firmly, I just put down popular formulate without proof it is terms out the title as upper bound on the bit error probability. It turns out that terms out that the generating function a end L I D derived earlier can be used to provide an upper bound on the probability of bit error incurred while employing the Viterbi decoder

(Refer Slide Time: 47:05)



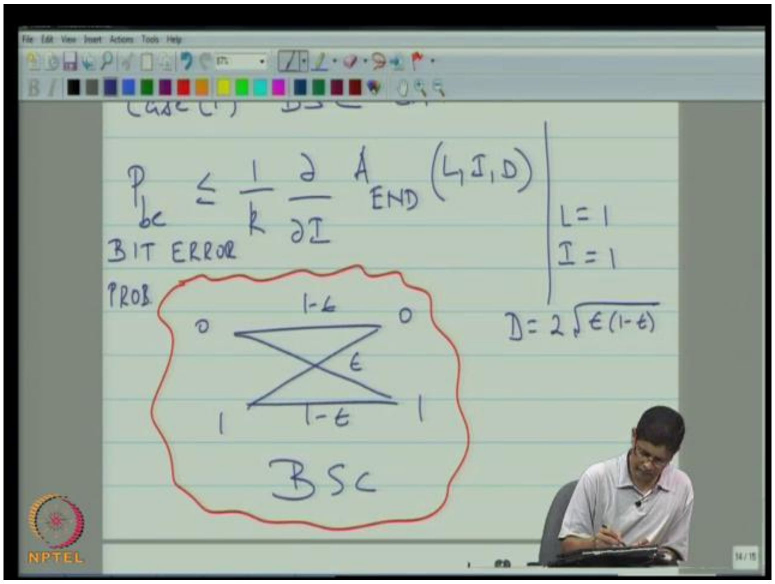
Now, just note here Viterbi decoder is actually designed to select that code word whose which is which minimize the probability of code word error, so in some of strange actually say that is basis for the basis for decoding. I will meant yet when we talk about performance analysis in terms of probability of error, we talk about bit error probability the basis weight in turns of batch convenient to this. So, that bit error an upper bound in the bit error probability is convenient derived and is of the great element, because bit error probability is better reflection of performance of code it is better and it is feasible.

(Refer Slide Time: 50:12)



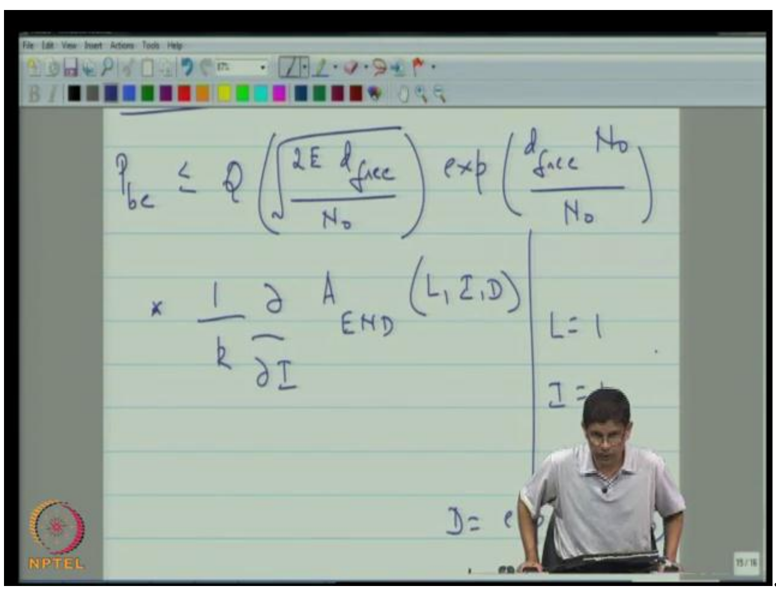
That is easy I do it and so again possibility we do not have time goes through do the derivation at take the selection to much or way. So I just put down the formula leave it like this there are 2 different formulas depending upon whether looking at band width symmetric channel or the channel. Case one for in the case of band width symmetric channel we have that probability of bit error is less or than equal to 1 upon k times del by del i of a end L I D, where a said after differentiating l equal to one i is equal to 1, and you said D is equal to twice root epsilon in to 1 minus epsilon and there is enlarge the formula.

(Refer Slide Time: 51:12)



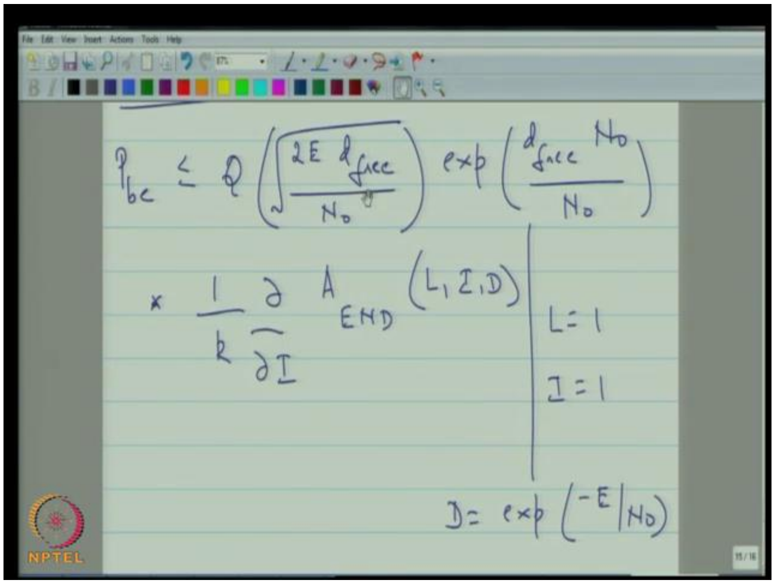
So here of course epsilon is the cross over probability on the band width symmetric channel right. So here we talking about, so this is sorting for this and of course here what we mean by this, this is reference to the bit error probability by bit means message bit error probability.

(Refer Slide Time: 52:31)



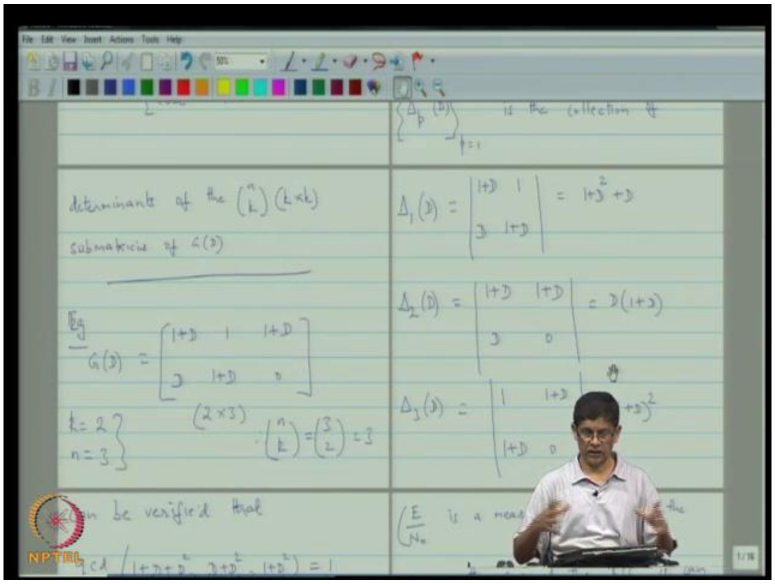
It take with derivative of n f i D with respect to i n is certainly equal to 1, i equal to 1 n D equal to 2 times root of epsilon 1 minus epsilon of course. Since a 1 is since l is not involved derivation you can actually certainly one to begin with that is alright, then case 2 in the case of the channel in the probability of error expression is a little more complicated in terms of write it down. This is it is in terms of k function. So expression is more I like it more complicated but just batch the way of partial explanation the heart of the formula really lies in this and decode actually this except example realise that decode actually make it little back title if actually wrote it this form.

(Refer Slide Time: 54:00)



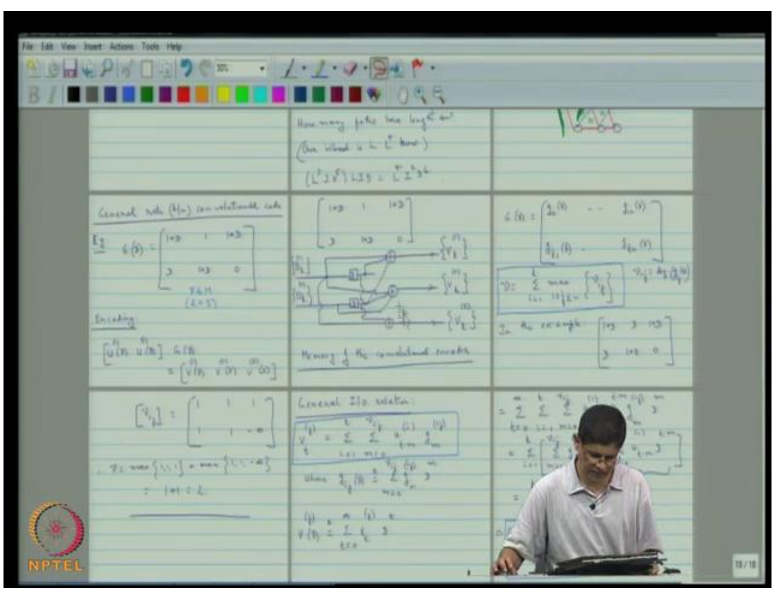
Now only before, I close I should just mention, perhaps let me just summarize what we done way, let me zoom out that you can take look at our slides here. So we today, I first of all wrote down, so today was kind of cataloguing clause when we actually write down, list on all topics in effort to try to provide somewhat complete discussion on convolution codes. As I mentioned earlier, since coding theory has a branched out and develop so many methods.

(Refer Slide Time: 54:49)



We do not have time to precede this in the depth that we could earlier. So for this reason this is the little bit rushed, so what we did was we actually derive wrote down the condition to avoid coming up to the polynomial generated matrix that has catastrophic error propagation and the condition was in terms of determinant of sub matrices and their gcd. And then after that we actually, talked about Viterbi decoding of additive quasi noise channel; and we went through an example there is not completely but it.

(Refer Slide Time: 56:02)



So for sufficiently to give you an idea what it is like. And then I wrote down expression for bit error probability of the convolutions code, and viterbi decoding. The only thing that perhaps remains to be done is to explain what D free is I think that so I will say quickly in words and may be comedown back to this next time. So when we in the last class, when we wrote down this generating function here, so let me zooming in to this point here, when we wrote down the generating function here, then when you expand this in a power series, the smallest exponent of D is called D free that perhaps I will say, few words about that next time and then begin in next topic.